IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 30, 2022, accepted April 5, 2022, date of publication April 20, 2022, date of current version May 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168972

SDN Security Review: Threat Taxonomy, Implications, and Open Challenges

MOHAMED RAHOUTI[1], (Member, IEEE), KAIQI XIONG[2], (Senior Member, IEEE), YUFENG XIN[3], (Member, IEEE), SENTHIL KUMAR JAGATHEESAPERUMAL[4], MOUSSA AYYASH[5], (Senior Member, IEEE), AND MALIHA SHAHEED[1]

[1]Department of Computer and Information Science, Fordham University, Bronx, NY 10458, USA
[2]Cyber Florida, University of South Florida, Tampa, FL 33620, USA
[3]RENCI, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
[4]Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu 626005, India
[5]Department of Computing, Information, and Mathematical Sciences and Technology, Chicago State University, Chicago, IL 60628, USA

Corresponding author: Mohamed Rahouti (mrahouti@fordham.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant 1633978, Grant 1620871, Grant 1620862, Grant 1636622, and Grant 1531099; in part by Raytheon BBN Technologies (BBN)/GENI Project Office (GPO) through NSF/Division of Computer and Network Systems (CNS) Grant under Project 1936; and in part by the Florida Center for Cybersecurity (Cyber Florida) for a Seed Grant.

ABSTRACT Software-Defined networking (SDN) is a networking paradigm to enable dynamic, flexible, and programmatically efficient configuration of networks to revolutionize network control and management via separation of the control plane and data plane. The SDN technology has evolved in response to the demands from large data centers toward all types of networks, from IoT, enterprise, to ISP networks. On the one hand, SDN has provided solutions for high-demand resources, managing unpredictable data traffic patterns, and rapid network reconfiguration. It is further used to enhance network virtualization and security. On the other hand, SDN is still subject to many traditional network security threats. It also introduces new security vulnerabilities, primarily due to its logically centralized control plane infrastructure and functions. In this paper, we conduct a comprehensive survey on the core functionality of SDN from the perspective of secure communication infrastructure at different scales. A specific focus is put forward to address the challenges in securing SDN-based communications, with efforts taken up to address them. We further categorize the appropriate solutions for specific threats at each layer of SDN infrastructures. Lastly, security implications and future research trends are highlighted to provide insights for future research.

INDEX TERMS Software defined networks, OpenFlow, security, threat, attack, vulnerability, network security.

I. INTRODUCTION

The Software-Defined Networking (SDN) infrastructure primarily aims to make networks programmable, thereby supporting them with higher flexibility and agility in terms of configuration, monitoring, and performance [1]. Before the evolution of SDN, data centers possessed challenges in coping with unpredictable data traffic patterns. Those data traffic patterns cause higher demands for resources that could not be met with conventional network infrastructures. The data center management experts had two choices: i) Scale up the network infrastructure, which is very expensive, and the majority of the networks are underutilized most of the time. ii) The Build a reconfigurable network that could automatically cope with the demands of the users and allocate computing resources to meet their appropriate demands. This is where the SDNs come to their rescue as a powerful tool for customizing the networks on the fly. In addition to providing various security supports, as shown in Figure 1, SDN enhances networking control functions for routing and policy definitions in an automated way.

SDN provides a new level of programmability and abstraction to a network layer, which plays a leading role in automating the network. SDN ensures network reachability at a faster rate by addressing the challenges in IP allocation, routing changes, bandwidth allocation, and policy openings. SDN allows system administrators to manage and control their network dynamically. Further, SDN infrastructures are highly

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

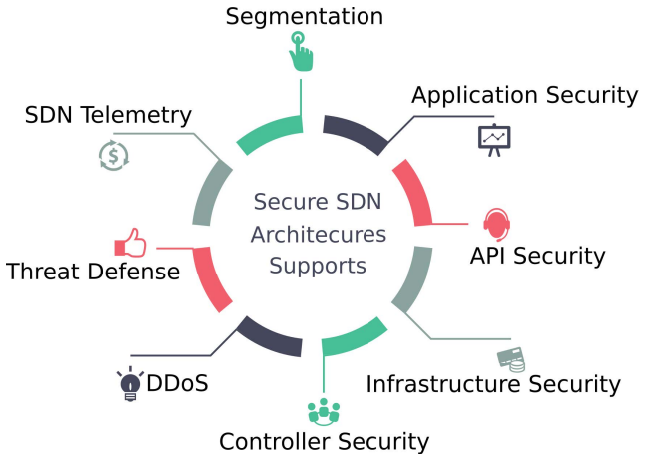


FIGURE 1. Security support provided by secure SDN technology.

correlated with Network Function Virtualization (NFV), enabling the building of virtual platforms [2]. They help to virtualize networking functions by allowing specialized hardware to be replaced with software applications that could run on universal devices. Efficient management of Internet of Things (IoT) security threats with the aid of SDN and NFV are reviewed by Farris *et al.* [3] for enhancing and addressing the cybersecurity challenges in IoT systems. Security issues in the control plane of SDN are addressed by comparisons with the conventional network models [4]. The authors also developed effective consistency checks and incorporated fault tolerance in their proposed SDN security architecture for the control plane. Another potential research prospect of SDN security is meeting its Quality of Service (QoS) requirements and dealing with heterogeneous networks [5], [6]. Authors in [7] developed a framework for transforming the SDN controllers into homogeneous groups and enhanced their security concerns by maintaining the robust security features in the SDN. Encryption schemes are developed for addressing the trade-off between the link security and communication performance [8]. Its effectiveness and validity are tested on SDN/OpenFlow platform in the security gateways.

The widespread adoption of SDNs is also prone to challenges in providing authentication, data privacy, and access control in networking routing. Sallam *et al.* [9] proposed a Software-Defined Perimeter (SDP) framework for restricting unauthorized access to the network as well as establishing an orchestration of network connections. Such integration of SDP with SDN enables handling bandwidth attacks through proper Denial of Service (DoS) and it could also be examined through virtual network testbeds. Integration of SDN and blockchain for handling cyberattacks in Industrial IoT (IIoT) devices are proposed by authors in [10]. They focused on developing a security architecture comprising intrusion detection system and integrity checking strategies using blockchain, thereby avoiding tampering of SDN-based IIoT systems. Authors in [11] exploited the capabilities of SDN with trusted agents for detection and denial of Rogue Access Points (RAPs) by utilizing hidden subnets, a network emulator, and a modular programming platform.

As an extension of the popular INET framework, Tiloca *et al.* [12] simulated SDN-based anomaly detection. They have also evaluated the performance of the architecture for managing the cyber-physical attacks in SDN and performed design optimization based on the analysis. IoT devices are most prone to attacks, and attackers easily target such devices unless they are secured and access by unauthorized hackers is restricted. Matheu *et al.* [13] developed the Manufacturer Usage Description (MUD) model for security policy enforcement with the aid of SDN techniques. This technique helps to protect the devices shared through blockchain-based platforms.

Abdou *et al.* [14] have made a comparative security analysis of control plane security in SDN and other conventional networks. Their analysis strategy supports fault tolerance and checks on consistency issues as well as the potential of applying robust security features in SDN systems. Zhong *et al.* [15] addressed load balancing and failure recovery approaches in SDN using a smart cooperative platform. While this secure communication mechanism is enforced using the cooperative platform and distributed controllers, the effectiveness of the authentication code plays a significant role. Distributed denial of service (DDoS) is one of the common issues that may occur in networks that lack robust security features. Prior research has examined its impact to a larger extent. Furthermore, the impacts of DDoS with its security implications in SDN frameworks are highlighted by the authors in [16] along with the recent research trends.

Furthermore, security becomes more critical in the underlying SDN infrastructure and the rapid increase in the number of smart devices connected to the SDN networks has not only increased the data traffic in the network, but also raised concerns on security aspects of SDN communications. For instance, SDN has been used as a backbone framework to effectively address and protect the information and communication infrastructure in ships, reported as CyberShip-IoT in [17]. The authors used a translation mechanism and higher-level policy language of higher level for evaluating the security metrics. Here, the performance is validated on different use cases. SDN-based vehicular networks are gaining momentum as key enablers with the support of 5G services to establish Intelligent Transportation Systems (ITS). The security issues in such vehicular networks are addressed at all the layers of the SDN, providing effective transmission of data packets through the data plane [18]. Moreover, countermeasures to the security attack on SDN-based vehicular networks are well addressed in [19]. Routing protocols addressing the security issues in SDN-based vehicular networks are dealt with in [20], in which the latency and secure connections are analysed. Vehicles communicating with each other in ITS can be attacked by unauthorized vehicles in the network. So, vehicular ad hoc networks are integrated with SDN-based solutions for ensuring reliable vehicle-to-vehicle (V2V)

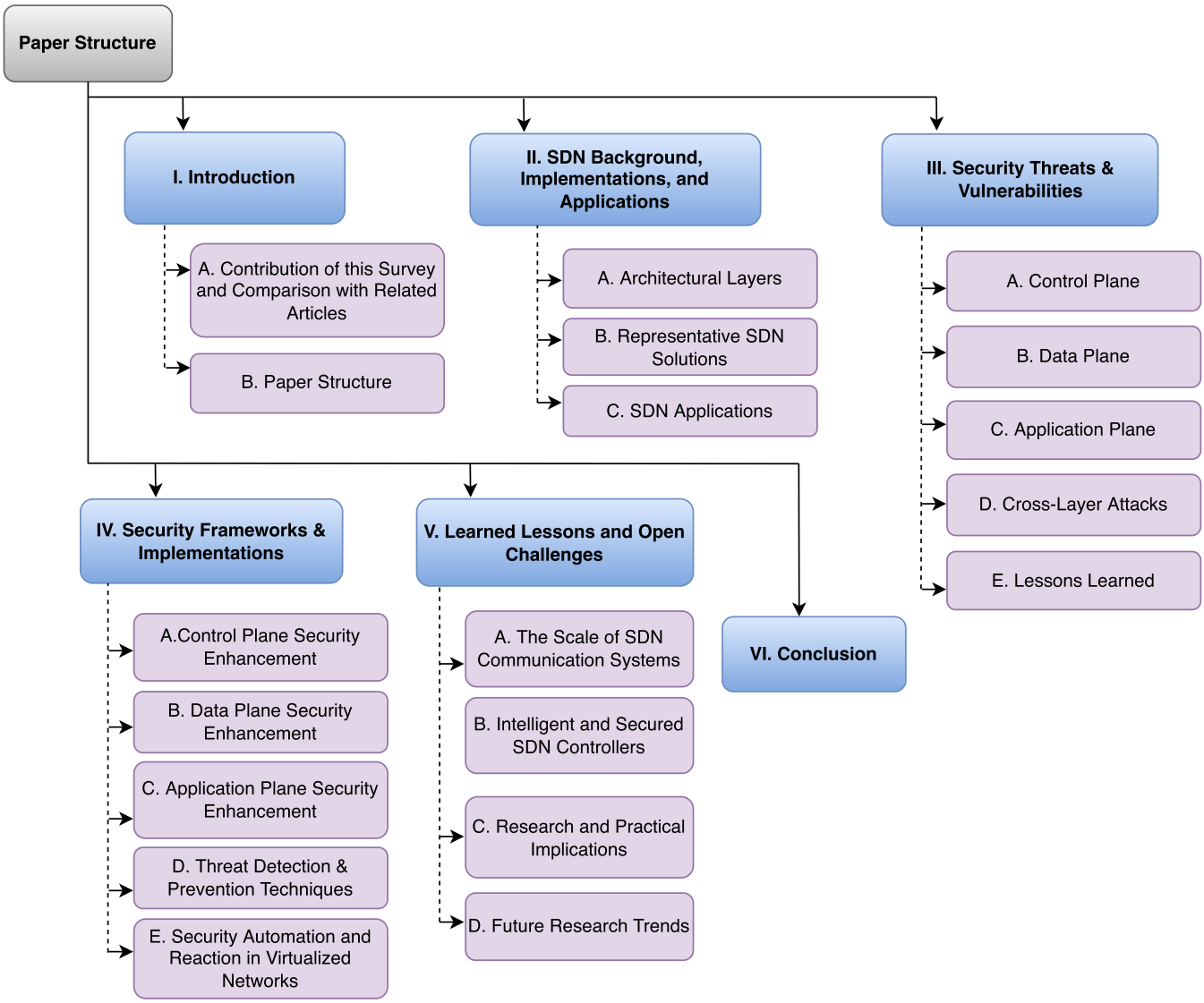


FIGURE 2. The paper's roadmap.

communication [21]. Here, through a single remote controller, the load on the network is controlled, and attacks on the vehicles are almost nullified.

Therefore, in this paper, we survey the core functionality of SDN from the perspective of secure communication infrastructure at different scales. A specific focus is put forward to address the challenges involved in securing SDN environments, with efforts taken up to address them. Building upon this, the appropriate solutions for specific threats at each layer of SDN infrastructure are reviewed along with their research and practical implications.

A. CONTRIBUTION OF THIS SURVEY AND COMPARISON WITH RELATED PAPERS

The objective of this paper is to provide a comprehensive view of the state-of-the-art security issues, security implications, and security enhancement practices in SDN communications. Contributions of this survey include highlighting the roles of the entities involved in SDN communication security, while integrating their corresponding application domains. This survey also discusses recent research efforts to solve, key security-related challenges in SDN communication and open-end research directions for future work. Table 1 shows comparisons between our paper and existing ones in the literature. Specifically, our paper contributes to the existing literature by adding an up-to-date review of security challenges in SDN infrastructure, discussion of current practices towards security enhancements per SDN plane, induction of research and practical implications of the reviewed security aspects. Although there are several variants of SDN technologies and protocols, in this paper, we mainly focus on OpenFlow, the most dominant SDN-enabled communications protocol. A summary of the contributions of this survey is enlisted as follows:

TABLE 1. Comparison of existing survey papers about SDN environment security. ✓, ✗, and ∗indicate that the topic is well covered, uncovered, and partially covered, respectively.

Ref.	Year	SDN Background	Use Cases	Security Review	Solutions Review	Detection & Mitigation	Implications	Research Directions
Scott-Hayward et al. [22]	2013	∗	✗	✓	✓	∗	✗	∗
Jarraya et al. [23]	2014	✓	✗	✗	✗	✗	✗	∗
Garg and Garg [24]	2014	∗	✗	∗	∗	✗	✗	∗
Farhady et al. [25]	2015	✓	✗	∗	✗	✗	✗	✓
Ahmad et al. [26]	2015	∗	✗	✓	✓	∗	✗	∗
Kreutz et al. [27]	2015	✓	✗	∗	✗	✗	✗	∗
Brian et al. [28]	2016	∗	✗	∗	✓	✗	✗	∗
Scott-Hayward et al. [29]	2016	✓	✗	✓	✓	✗	✗	✓
Masoudi and Ghaffari [30]	2016	✓	✗	∗	✗	✗	✗	∗
Dayal et al. [16]	2016	∗	✗	∗	∗	∗	✗	✗
Cox et al. [31]	2017	✓	∗	∗	✗	✗	✗	✓
Dargahi et al. [32]	2017	✓	✗	✓	∗	✗	✗	✗
Gharaibeh et al. [33]	2017	✓	✗	∗	∗	✗	✓	✓
Abdou et al. [14]	2018	∗	✗	∗	✓	✓	✗	∗
Sultana et al. [34]	2019	∗	✗	∗	✓	✓	✗	∗
Nisar et al. [35]	2020	∗	✗	✓	✓	∗	✗	∗
Shaghaghi et al. [36]	2020	∗	✗	✓	✓	✗	✗	∗
Hussein et al. [37]	2020	∗	✗	✓	✓	∗	✗	∗
Chica et al. [38]	2020	✓	✗	✓	∗	✗	✗	✓
Mazhar et al. [39]	2020	∗	✗	∗	∗	✓	✗	∗
Han et al. [40]	2020	✓	✗	∗	∗	✓	✗	∗
Yurekten and Demerci [41]	2021	∗	✗	∗	✓	✓	✗	✓
Our survey	2021	✓	✓	✓	✓	✓	✓	✓

- A discussion on the SDN architectural layers and their operations and functionalities.
- Motivations (with SDN use cases) towards awareness of the security attacks and vulnerabilities that affect the SDN infrastructure and communication.
- Review on the recently evolved SDN-enabled applications with security enhancements in emerging areas as use cases (e.g., Internet of Things, LTE and 5G communications, Blockchain technology, time sensitive networks, software defined WAN, etc.)
- An up-to-date review on the security challenges and threats in SDN infrastructure.
- Review of existing solutions and practices towards enhancing security in SDN infrastructures. The solutions are reviewed in accordance with SDN planes with a categorization of threat detection and mitigation techniques.
- Induction of research and practical implications by the reviewed security aspects.
- Discussion of the future research directions and open challenges hindering resiliency and security in SDN-enabled systems.

Different from existing papers, this paper provides an up-to-date discussion about security issues in SDN environments. The presented taxonomy covers security threats ranging from topology discovery, network and app manipulation, flow diversion, teleportation & rootkits, controller placement, access control and authorization, to conventional networking threats. Further, in addition to reviewing existing security solutions and related implications, this paper provides key use cases of SDN technology implementation in smart communication systems. This layout is aimed to enable researchers to focus on the challenges that breach security in SDN communications. Ultimately, this work will provide further insights for future research in the SDN security domain.

B. PAPER STRUCTURE

The rest of the paper structure is shown in Figure 2 and organized as follows. Section II provides a background of the SDN that is enabled by other supporting technologies that drive its implementation and integration towards end applications. Section III discusses the security attacks and vulnerabilities that do exist in the underlying SDN infrastructure and communication. Section IV further covers security challenges hindering security enhancement in SDN infrastructure. Section IV also reviews and classifies existing SDN-enabled solutions and implementations to thwart the different security threats in the underlying planes of SDN. Detection and mitigation techniques are also reviewed in Section IV. In Section V, learned lessons are induced in accordance with the presented review in this paper. Moreover, this Section V projects the research and practical implications of SDN security along with future research directions and open challenges in this area. Last, Section VI concludes the survey.

II. SDN BACKGROUND, IMPLEMENTATIONS, AND APPLICATIONS

In this section, we introduce the key features of SDN and discuss the related literature in detail. The main aspects of integrating SDN into use cases highlighted here include smart cities as one of the key technologies. SDN is an evolving paradigm of networking technology aiming at superseding the limitations of conventional (i.e., legacy) networks [1]. The primary worth of SDN lies in its capability to provide and assure coherent policy enforcement, network programmability, enhanced scalability, and holistic visibility via

centralized management. Table 2 summarizes the existing SDN controller software and their specifications. SDN infrastructural components are depicted in Figure 3 and their detailed descriptions are discussed next.

A. ARCHITECTURAL LAYERS

In this section, we characterize the architectural layers of SDN in terms of their roles and key functionalities that could be tweaked for a diversified range of use cases.

1) APPLICATION PLANE

The application plane handles the services and applications request network functions from the data and the control planes. In the standard network configurations, the monitoring and the control of devices occur in this layer [59]. Although the devices' functions are similar to the SDN networks, the delivery modes are often virtualized, centralized, and abstracted. Furthermore, the attributes, services, and rules are also defined in this layer, where network information about the device's topology and the appliances is needed to implement a broad range of end-to-end SDN-enabled services efficiently. Additionally, applications, through this layer, can make timely decisions depending on the network changes [60]. Overall, the scope of the application layer includes security enforcement and largely considers the access policies while providing load balancing, traffic engineering, network management, and monitoring [31], [40].

2) CONTROL PLANE

Unlike the traditional network architecture with integrated control and data planes, SDN enables a decoupled architecture to feature a distinct control plane that defines network management/control and traffic routing. The control plane is a layer dedicated to managing, monitoring, and configuring forwarding processes (e.g., flow forwarding decision) across the network stack [59]. The separation of control from the data plane in SDN enhances agility in the automation, monitoring, management, maintenance, extension, provision, and troubleshooting of network infrastructure [40]. Based upon a centralized control, the controller, via the control plane, sends necessary and appropriate information to the forwarding devices (e.g., OpenFlow switch) as flow rules for effective decision-making. This is achieved through explicit forwarding information base (FIB) and MAC programming [31], [40], [61].

3) DATA PLANE

SDN addresses the limitation of the network architecture by instituting flexibility in scaling and supporting dynamic computing environments as speed and efficiency decline with an increase in forwarding devices and connected devices in conventional networks. The SDN data plane executes programmable algorithms (controlled by the control plane) that coordinate and enhance management processes and device configurations. As a result, networks were capable of efficiently accommodating the dynamics of changing configurations without constraining the efficiency [62]. The data plane entails numerous network constituents that allow uninterrupted connectivity. This plane oversees the transfer of data flows between end hosts [31], [38].

The data plane also fosters network integration and interoperability across all traffic engineering (TE) processes, including device management and configuration [41]. In this context, the architecture provides neutrality for optimizing infrastructure investments to foster commercial and nominal requirements. Data plane provides standard protocols for instituting a typical software environment, where devices from different vendors can communicate. Moreover, it manages open and modular dimensions by standardizing network applications, systems, and policies [38]. Open network management or orchestration simplifies deployment, creating a highly scalable network with enhanced performance optimization capability.

B. REPRESENTATIVE SDN SOLUTIONS

1) OpenFlow

OpenFlow is an open interface protocol for remotely controlling the forwarding tables in the network switches, routers, and access points. By setting up the virtual machines tools and providing the hubs as Ethernet switches, the OpenFlow helps modify the network flows and enables multiple switch support. Further, it enables the switch to handle IP forwarding, run the switch on a real network as well as add firewall capabilities to the switch [63]. According to OpenFlow protocol specifications, security aspects need to be incorporated over the SDN control layer, and the SDN controller must grant a network view that is unified and clear to render security threats and violations, which are easy to spot as well as ensure security policies installment [64]. Further, the OpenFlow protocol are also bpound to few limitations, including the disallowance to choose specific functions (when not all functions are needed), and a new OpenFlow version typically requires hardware with rich functionalities [65].

2) SEGMENT ROUTING (SR)

SR is a data plane technology that provides a control solution for the SDN flow path. SR is simpler and more scalable than conventional SDN protocols such as OpenFlow. Security in SDN can be well-comprehended with SR. It chooses an optimized path and encodes it in the packet's header, and transmits as an ordered sequence of segments. The segment holds identification for instructions such as context, locator, and services. These services can be leveraged through multiprotocol label switching (MPLS) in conjunction with the Internet protocol for listing the segments in the routing extension header [66].

Further protocols and solutions for traffic engineering and management that can be integrated into SDN technology are summarized as follows:

- Path Computation Element (PCE): PCE is a controller that computes the network's paths. By applying

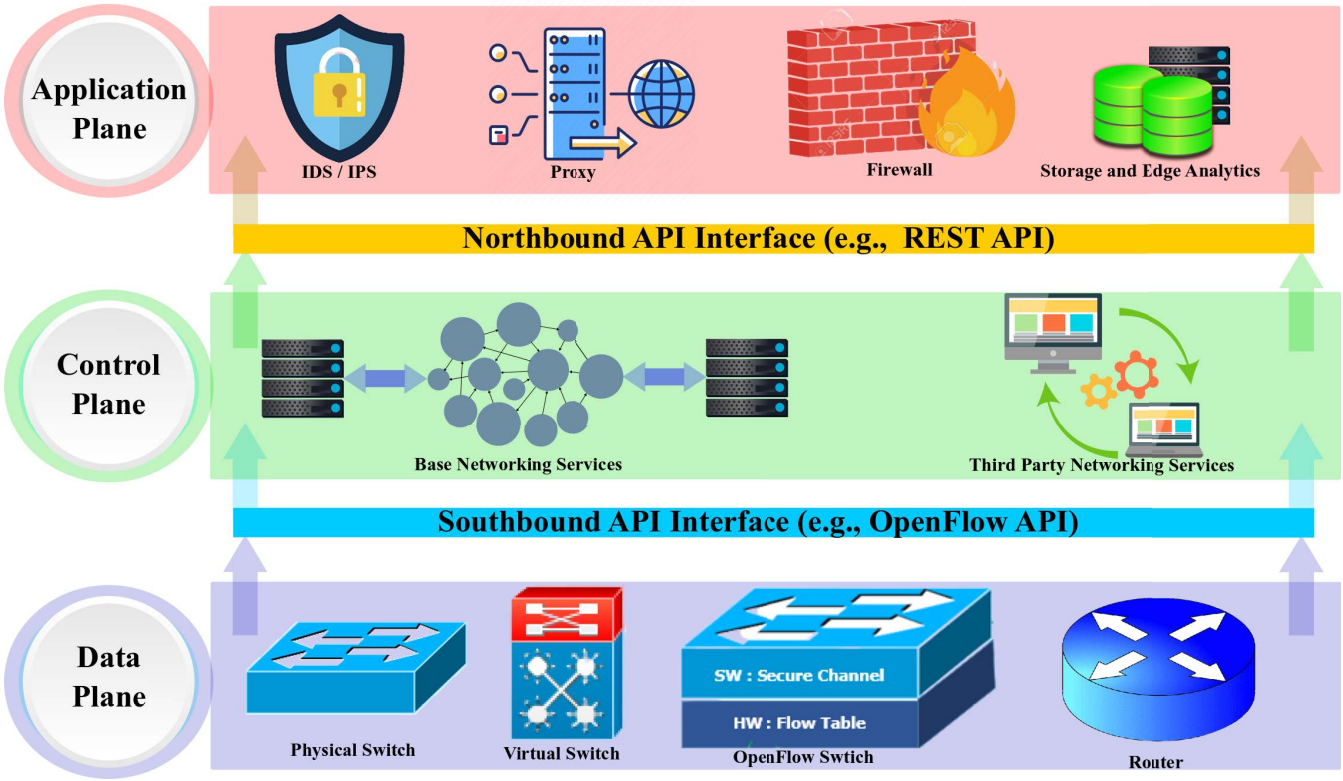


FIGURE 3. The layers of SDN infrastructure and its components.

TABLE 2. Existing SDN controller software.

Controller	Language	Physically Distributed	Multi-Threaded	TLS	Rest API	% Deploying	Other Features
Floodlight [42]	Java	✗	✓	✓	✓	11	GUI, forked from Beacon
ONOS [43]	Java	✓	✓	-	-	23	Built for service providers, supports OVSDB, BGP, Nteconf, and TLI
OpenDayLight [44]	Python/Java	✓	✓	✓	✓	61	GUI
NOX [45]	C++	✗	✗	-	-	-	Deprecated
Ryu [46]	Python	✗	✗	✓	-	15	Frequent switch certifications
SNAC [47]	C++	✗	✗	-	-	-	Built on NOX, GUI, closed source
HyperFlow [48]	C++	✓	✓	-	-	-	Built on NOX
OpenMUL [49]	Python	✓	✓	✓	✓	8	Supports Netconf and OVSDB, stable performance
Kandoo [50]	Go	✓	✓	-	-	-	-
OpenContrail [51]	Java/Python	✓	✓	✓	✓	14	Compatible with OpenStack. Southbound API: XMPP, BGP and Netconf
Trema [52]	C/Ruby	✗	-	-	-	-	-
Beacon [53]	Java	✗	✓	-	-	-	GUI, limited to STAR topology
POX [54]	Python	✗	✗	-	-	-	GUI, Development stagnated
Ryuretic [55]	Pyhton	✗	✗	✓	-	-	Built on Ryu
DIFANE [56]	C	✓	✓	-	-	-	Built on NOX
Pyretic [57]	Python	✗	✗	-	-	-	Deprecated built on POX
OPNFV [58]	Python/Java	-	✓	✓	✓	26	Compatible with NV/SDN, ODL, OpenContrail, and ONOS

computational constraints, the PCE can compute the optimized path on the network graph with enhanced scalability. Stateless PCE is capable of path computations in response to path requests from path computation clients (PCC). Further, PCE is highly appreciable to be deployed in scalable and resilient network traffic management. In [67], one such instance was reported with secure means of Erlang-based traffic engineering for SDN deployed for robust implications in 5G services with increased scalability of the control plane. The valuable inclusion of PCE further aids in providing the optimized path and assists in interacting with applications for making dynamic changes in SDN environments through PCE.

- Multiprotocol Label Switching (MPLS): It is an architectural framework that decouples services from the transport layer, where the decoupling is accomplished via encoding instructions in the packet header. In alignment with the continuous evolution of OpenFlow, MPLS is also gaining momentum as an SDN-enabler protocol. The MPLS paradigm (decoupling service from transport, placing instructions on packet headers) has recently gained more relevance in the deployment and development of scalable SDN communications (SDN 2.0) [68].
- Abstraction & Control of Transport Networks (ACTN): They are involved in creating abstract topology for each application or based on the demand from each user. This multi-domain network topology is responsible for addressing a diversified range of customers through its physical network control that can be implemented with the services offered by PCE [69]. Such frameworks enable logical application overlays and abstraction in the management of network traffic with optimized control plane disaggregation and simplify the SDN controller's operations.
- Interface to Routing System (I2RS): I2RS agents in the network router are capable of coordinating with the topology, events, policies, I2RS clients, as well as routing and signaling protocols [70]. The data encoding mechanisms implemented in the I2RS agents aim to provide secure use cases, predominantly applied for traffic steering, traffic classification, DDoS mitigation, topology control, and service chaining. It offers a unified interfacing solution to the routing system in order to achieve better monitoring and routes for data transmission in SDN networks.

3) PROGRAMMING PROTOCOL-INDEPENDENT PACKET PROCESSORS (P4)

P4, being one of the domain-specific programming languages, specifies the process of forwarding the packets by the data plane. SDN control protocols like OpenFlow could be operated in conjugation with the P4 for imparting reconfigurability, make the switches operate independently of the vendor, and enable programmers to describe the processing functionality independent of the target and underlying hardware [71]. At present, the capabilities of P4 could also be leveraged to program end-to-end multi-layer information in IP-enabled optical networks for provisioning flexible multi-layer in-band network telemetry [72].

4) APPLICATION DRIVEN SDN

- Application Layer Traffic Optimization (ALTO): ALTO protocol provides abstract and useful network information and ensures efficient network usage with better traffic optimization. It is developed on top of existing REST APIs [73]. Moreover, its service-aware parameters are highly useful for real-time communication, live media streaming, and peer-to-peer (P2P) file sharing. Indeed, the ALTO could also be integrated with PCE for establishing data center interconnect, which ensures joint optimization of resources. Further, through a meta-heuristic clustering mechanism with coordination of ALTO [74], they could be helpful in the implementation of dynamic frameworks for the detection of DDoS in the SDN services.
- REST API: For establishing communication with the network switches, REST API (aka RESTful API) is one of the popular choices over OpenFlow. The controller applications with the REST modules supports towards the insertion of flow entries and flow tables in the SDN switches. REST design approaches assists in addressing the challenges normally encountered in the northbound API of SDN. The truly RESTful approach makes the SDN a pure service-oriented data networking platform. Further, in an SDN-enabled cloud environment, REST API access control mechanisms are sensitive to pave the way towards the provision of secure application management in the prescribed framework [75].

C. SDN APPLICATIONS

To realize the benefits of SDN in different applications, efficient implementation of technologies such as IoT, wireless communication, blockchain, etc., are key requirements. This section outlines the core concepts under the aforementioned supporting services for SDN integrated smart applications.

1) DATA CENTERS

In an enterprise, by placing computing power closer to the sources of data, such as routers, WAN, and IoT devices, the use of edge computing makes the data travel less distance compared to a distance cloud infrastructure. Edge computing eases the strain on resources by processing data close to the source and only sending valuable data to the cloud services. Computing terminals demanding reduced response time, conserving network bandwidths, and reduced data bottlenecks could be reasonably addressed using edge computing. However, as it's a distributed architecture, they are prone to security challenges and opens up for possible attacks. Reasonable trade-offs in choosing cloud or edge computing with the support of SDN frameworks could address the security threats on such computing. In [76], improved Quality of Experience (QoE) is guaranteed with the support of SDN that provides dynamic configuration and mobile edge orchestration of the network with enhanced services and the security aspects imparted through network intelligence.

Data security and privacy issues in a pervasive edge computing paradigm can be addressed through SDN-enabled blockchain services, which could provide secure device-to-device communication with flexible operations [77]. The work by Li *et al.* [78] demonstrated the means of addressing the security issues in IoT-enabled healthcare systems using a secured framework implemented using SDN-based edge computing services. Here, the simulations highlight the load balancing and network optimizations authenticated through the edge servers connected to the SDN controllers. Recently,

SDN-based mobile edge computing was proposed in [79], where the authors focused on dynamic service migration and joint edge caching through deep Q-learning. This approach considerably reduces the transmission cost and several other service migrations.

In most healthcare and industrial networks, real-time critical communication is highly relevant for forwarding vital data for processing. At present, however, the devices are becoming more intelligent and transmitting not only real-time capable data, but also other data for predictive maintenance and energy optimization. Time-Sensitive Networking (TSN) ensures delivery of real-time relevant data in the network, even despite increased data load. They can be used to control data communication in the form of time synchronization and prioritization of data streams. This ensures that an application neither interferes with the other communication nor is disturbed by them. For time-sensitive industrial IoT applications, the authors in [80] proposed an SDN architecture based on online strategies. It ensures optimized routing, scheduling, and admission control along with the guaranteed allocation of secure transmission time-slots.

Further, the work in [81] focuses on failure handling for time-Sensitive networks using SDN and source routing, which are evaluated by integrating into Mininet using Linux-based TSN scheduling. In another recent work by Kong *et al.* [82], failure analysis of time-triggered traffic and its run time recovery is demonstrated in time-sensitive networks. IHSF [83] was proposed as an intelligent solution for providing reliable and time-sensitive flows in hybrid SDN-based IoT systems, which ensures improved performance and secured means of data communication. Also, it provides a better end-to-end delay in terms of optimized network observability time, with an excellent packet delivery ratio. In [84], SDN-based self-configuration strategies for time-sensitive IoT networks are developed for different numbers of end-hosts and network sizes. Such strategies provide guaranteed and secured means of real-time data transfer without prior knowledge of the flow rates.

2) SOFTWARE DEFINED WIDE AREA NETWORK (SD-WAN)

The ultimate objective of Wide Area Network (WAN) is to connect users to their applications from anywhere across the globe anytime, and from any device with the appropriate interface. In most cases, it adds up delay that degrades application performance, and it consumes costly leased line bandwidth. More intelligent ways of reaching the application through the Internet are driven by the software-defined model for the WAN. Instead of routing traffic solely based on addresses, an SD-WAN is an application-aware access mechanism that uses software to more intelligently route or steer traffic across the WAN based on the business requirement for an application in a secure manner. The SD-WAN has revolutionized WAN services through application-driven networking to meet the demands of customers and services. Usually, this makes a close connection to the security and cloud computing for easy management of the resource regardless of the connectivity provider [85]. IoTSim-SDWAN [86] was developed as a simulation framework for interconnecting distributed data centers over SD-WAN for dynamic computation of the best and secure route for the network flow. Further, this work proposes a coordination scheme to link the SDN controllers and the SD-WAN.

Further, energy and network performance issues were evaluated using the IoTSim-SDWAN simulator and compared with the classical WAN services. The authors in [87] evaluated the dynamic traffic management for SD-WAN intercloud communication. It minimizes the internal linkages, ensures secure communication, and reduces traffic costs by operating in a distributed manner. Further, SD-WAN has found its application in healthcare services as one of the recent works reported in [88]. Here, with SD-WAN as a backbone network, the healthcare systems are evaluated in terms of latency, jitters, communication bandwidth, security, privacy, and trust in the systems to ensure their reliable services.

3) WIRELESS COMMUNICATIONS: LTE AND 5G

The role of SDN in LTE and 5G makes the network more agile and flexible by carefully designing, managing the networks, and programming the control plane. SDN further provides an intelligent architecture for LTE and 5G networks. It also plays a crucial role in creating multiple network hierarchies and intelligent frameworks for LTE and 5G network programmability. For integrating the SDN with the LTE and 5G services, the forwarding device with embedded control will be linked with the firewall and the traditional network with distributed cloud services for establishing secured means of communication. An SDN controller driven through the software control establishes the communication link between forwarding devices and decoupled network control. The authors in [89] demonstrated the performance of aggregated LTE and WiFi with open-source frameworks by testing them on UDP and TCP services, considering three types of policies for securing the communication in the network. Further, in [90], the authors used blockchain for building secure and trustworthy IoT services in SDN-enabled 5G vehicular ad-hoc networks (VANETs).

4) SMART CITY

a: INTERNET OF THINGS (IoT)

Some predominant communication and security issues in IoT can be addressed with the support of SDN technology, thereby assisting in making efficient IoT services. Integrating SDN with IoT enables the deployment of intelligent routing decisions for IoT communications [91]. It also simplifies the information collection mechanisms and helps to make better analysis and decision making [92]. Further, the network management services can be simplified based on the IoT device, user specifications, application-specific requirements, and enhanced visibility of network resources are guaranteed using SDN technology. While the IoT devices communicate with

other devices or cloud services, the SDN delivers intelligent traffic pattern analysis and coordinated decision-making.

b: SMART GRIDS AND POWER GRIDS

The ever-changing and rising energy demands for household and industrial appliances require the implications of smart power grids. They can introduce a two-way dialogue to exchange electricity and information between the customers and the utility service providers. Smart grids have evolved as a developing network of communications, control, and automation with the support of new technologies like SDN, working together to make the grids more efficient, reliable, and more secure [93]. These smart grids assists in the integration of renewable energy sources such as solar and wind and are also capable of plug-in electric vehicle charging. A security framework for SDN-enabled smart power grids is developed by the authors in [94], to provide a robust and secure smart grid architecture. It is experimented on multiple SDN controllers with lightweight identity-based cryptography to protect smart grid networks from external attacks. The authors in [95] introduced the best features of SDN for supervisory control and data acquisition (SCADA) networks deployed in smart grid applications. Here, the resiliency-aware SDN, plans and executes the best alternate paths during network congestion and addresses the threat issues in the network. Recently, Mahmood *et al.* [96] demonstrated the effectiveness of an SDN-based DDoS protection system for smart grids by employing a lightweight defense mechanism. Further, the performance measures such as detection rate, resource utilization, DDoS protection focusing on various attacks are evaluated.

c: CONNECTED AND AUTONOMOUS VEHICLES (CAVs)

Through connected services and protocols, autonomous vehicles can talk to each other and transform the current transportation by enabling safe and interoperable networked wireless communications among them. Connected vehicles provide drivers with 360-degree awareness of the environment and keep personal information private. The drivers will receive warnings and will be informed about potential hazards through a visual display, seat vibration, or tones. The role of SDN in such use cases is supported through the SDN controllers to provision better resource management and resource allocation for the autonomous vehicles [97]. Here, multi-access edge computing along with SDN enhances resource utilization and resource management. MobQoS [98] is the mobility aware and QoS-driven SDN framework for autonomous vehicles, which incorporates distributed mobility management and provides improved routing decisions in the network. Further, this approach facilitates mobility-aware SDN frameworks for autonomous vehicles addressing the QoS challenges. In [99], the authors proposed an energy-efficient SDN controller placement strategy for SDN-based connected autonomous vehicles. Here, an efficient composite architecture is used to manage the underlying SDN communications by deploying multiple controllers.

d: ROBOTICS AND SMART MANUFACTURING

Robots used for smart manufacturing tasks are designed to carry out specific repetitive processes with minimum human intervention and high degree of precision. Robots were employed to save labor, enhance productivity, and avoid human error that may cost millions in smart manufacturing. Further, autonomous guided vehicles are also getting increasingly popular in industrial automation tasks. Establishing communication among the machines/robots and collaborating on their tasks are centralized challenges faced by industrialists and researchers. In [100], the authors proposed a dedicated communication channel using SDN controlled visible light communication to collaborate the tasks among multiple autonomous guided vehicles in a secure way. This approach considerably reduces the communication overhead and handovers, and the SDN paradigm makes communication much more secure for collaborative tasks among robots. SMOTE [101] is employed as an SDN-based multi-objective traffic engineering approach for performing telesurgery with the aid of robots. It was deployed to provide secure remote access to surgeons with better QoE. In [102], the authors used mobile edge cloud-based industrial IoT to improve the edge intelligence with the support of hierarchical SDN controllers. The authors proposed distributed and centralized control schemes to enhance the security aspects of smart industries' with improved edge intelligence.

e: BLOCKCHAIN TECHNOLOGY

As SDN communication is evolving from proprietary hardware to virtual software to ensure effective management of security in the networks, developers have started depending on the advances of blockchain technology. A blockchain-based security framework is developed by the authors in [90] to provide a secure and trustworthy IoT infrastructure in SDN-enabled 5G vehicular ad-hoc networks. Here, the blockchain-enabled scheduling procedures arrest the entry of malicious nodes that could be largely used in a vehicular IoT environment providing secure and trustworthy communication through the incorporated SDN architecture. Further, in another work by Derhab *et al.* [10], a blockchain-based integrity checking system is incorporated into the backbone of SDN-enabled industrial IoT systems. This system defends industrial IoT devices against cyber-attacks and provides an effective security solution. As energy-efficient SDN controller are of primary importance in large scale deployment, their security aspects in delivering enhanced services could be efficiently addressed through blockchain services [103]. Here, an efficient authentication method is incorporated through distributed trust services, ensuring lower energy consumption with lower delay and improved throughput in the routing protocol.

In the edge cloud infrastructure for IoT services, the incorporation of a blockchain-enabled security framework is addressed in [104]. In this framework, the dynamic network traffic flow management, which means a root cause for the

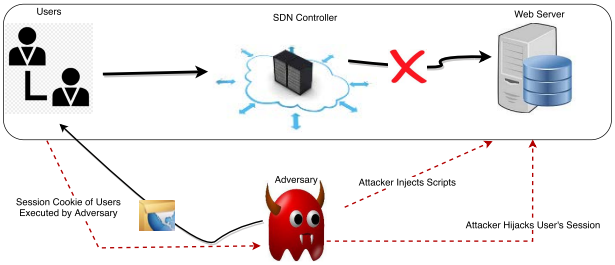


FIGURE 4. Host location-based hijacking attack.

TABLE 3. Host profile in different OpenFlow controllers.

Host Profile	SDN Controller
Location, IP, **MAC**, **VLAN ID**	Floodlight [42]
Location, **IP**, **MAC**, VLAN ID	Beacon [53]
Location, IP, **MAC**	POX [54]
Location, IP, **MAC**, **VLAN ID**	OpenDayLight [44]
Location, **IP**, MAC	RYU [46]
Location, IP, **MAC**, **VLAN ID**	Maestro [114]
Location, **MAC**	NOX [45]
Location, **MAC**	OpenIRIS [115]

security attacks, is recognized through the hindering doubtful flows. Further, data confidentiality issues are carefully addressed through the integration of blockchain in the SDN and the edge cloud services. SmartBlock-SDN [105] is developed as an optimized blockchain-based SDN framework for resource management in the IoT ecosystem. Here, a layered hierarchical architecture is presented, including a blockchain-enabled SDN-IoT framework providing an efficient selection of cluster-head and ensuring secure communication in the SDN network via the isolation of rouge switches through proper identification. The blockchain technology can further be leveraged to enforce tamper resistance and integrity of SDN data such as log files for forensics purposes [106].

Further, substantial hardware differences, communication standards disparity, and vendor-based software specifications restrain the successful attainment of Information and Communication Technologies (ICTs) and its services. Fortunately, the virtualization and softwarization progress in the transportation and network layers, in particular, can resolve some of such challenges. Some of the key softwarization technologies include SDN, NFV, and cloud computing [107], [108]. These softwarization enabling technologies are leveraged to integrate smart devices into SDN infrastructure and simplify information management in the underlying communication network [109]. Furthermore, NFV and SDN can jointly enable various data management services, including all the L2-L7 services and applications [110]. While several research studies focused on traffic engineering enhancement and service provisioning in SDN using NFV features [111], [112], NFV and SDN can also be leveraged to implement location-aware network virtualization and reconfiguration for the LTE networks [113].

III. SECURITY THREATS AND VULNERABILITIES

This section highlights the security threats and vulnerabilities in the emerging SDN technology, aiming to provide secure manipulation of the network, topology, flow rules, and access mechanisms. The security attacks and their implications are discussed and analyzed for each SDN layers.

A. CONTROL PLANE

1) HOST LOCATION HIJACKING

Resources allotted for the controller, particularly in the data to control plane, are prone to host location hijacking attacks. The attacker involved in this task typically slows down the network by consuming the resources allotted for the controller or even making the network inaccessible for the end-user. In SDN architecture, the intelligence of the whole network state and resources management reposes behind the centralized controller that could be a single point of failure in some cases, making it feasible to bring the entire networking system down from the adversaries' point of view by saturating the SDN control plane. This basically could establish a DDoS/DoS threat on the SDN data layer by utilizing host location hijacking attacks [116] as depicted in Figure 4. Furthermore, in the current implementations of different SDN controllers, there are few security restrictions on host location updates. Therefore, adversaries may easily impersonate any network identity with its index of host profile (MAC address) as shown in Table 3.

In each layer of SDN, the possible threats due to topology discovery are highlighted by the authors in [117]. Further, with the centralized and decentralized SDN controller setup, the robustness of the learning model against location hijacking attacks is assessed [116]. Here, it is assessed to mitigate the attacks with less resource overhead. With the programmable capability and adaptability in the control plane of SDN, it is capable of mitigating most of the vulnerabilities in the SDN topology [118]. Multi-layered intrusion detection and protection strategies are developed by the authors in [119] to mitigate IP Spoofing, control plane saturation as well as location hijacking issues in SDN. Microgrids connected to the SDN network are prone to cyber-attacks based on their topology. Active synchronous detection methods developed by Li *et al.* [120] help safeguard the network of microgrids.

2) LINK FABRICATION

An attacker in SDN involved in introducing a new malicious link over the network to gain control over the traffic is known as link fabrication attack (LFA). Due to the scalability concerns of SDN and the complexity of the network, the determination of the origin of LFA is highly challenging. The authors in [121] performed a statistical analysis on link latencies by undergoing a vetting period. Also, in comparison with the baseline models, relay-based LFA are assessed. Based on the assessment, the results concluded that relying on userspace increases the latency with fewer samples than the kernel space. For the examination of LFA in SDN, Khan *et al.* [122] proposed a formal strategy with a case

study. Here, the SDN that utilizes Higher-Order Logic (HOL) is considered along with the entire system liability, dependencies, host vulnerabilities, and service information. This helps to analyze the performance metrics in the attack. A secure and efficient OpenFlow topology discovery protocol [123] requires minimum changes in the design of switches and eliminates vulnerabilities to a larger extent in the process of topology discovery. During the evaluation process, the Floodlight controller considerably reduces the topology discovery time compared to the original OpenFlow topology discovery protocol. It provides improved performance with reduced discovery time by several orders of magnitude.

Further, Huang *et al.* [124] developed a lightweight topology verification scheme for establishing efficient SDN topology discovery with a higher level of trustworthiness. Here, the security threat model for SDN controller link and host location verification strategies are carried out. Also, the implemented TrustTopo in the SDN controller (Floodlight) helps to secure the network against topology attacks.

3) PORT AMNESIA

Defense strategies on the topology attack are gaining popularity in SDN networks. Port amnesia-based attacks bypass the port labels, impersonating the hosts and becoming actively involved as a drop in network flow rates. The authors in [125] developed an effective defense against port amnesia with the extension to TopoGuard. This mechanism helps the SDN to be more resilient against such attacks. Further, with the accurate view of the SDN states, the classification of poisoning attacks is addressed in [126]. Also, more emphasis was given to their impacts than other attacks and port amnesia attacks' bypassing nature.

4) PORT PROBING

Port probing is also one of the prominent attacks on the SDN topology, in which the attacker sidesteps the mechanisms used in the guards. Moreover, it baffles the host location by making the host of the victim move to a new network location, leading to host hijacking attacks. In conjugation with the amnesia attacks, the authors in [126] also emphasized the impact of port probing in SDN frameworks. Moving Target Defense (MTD) introduced in [127] helps to prevent the SDN from inside and outside attacks. Here, the MTD mechanism helps to positively reduce the poisoning attacks by integrating MTD with the SDN environment by utilizing the virtual IP addresses of the hosts.

5) PERSONA HIJACKING

In the SDN network stack, the bindings of the layers are prone to attacks leading to Persona Hijacking. Such an adversarial attack fools the SDN infrastructure by trusting the attacker as the owner, significantly driving access to the network resources. Persona hijacking exploits the weakness in the identifier binding mechanisms of SDN and involves the IP takeover phase and flow poisoning phase in the network. With the experiments carried out using the OpenDaylight cluster, the impact of the Persona hijacking is evaluated [128]. In this hijacking, the attacker tends to increase the CPU consumption on the nodes in the cluster, causing inconsistency for a period when the events are flooded.

The Persona hijacking also leads to Worm-Hole Attack in the SDN, which triggers flow misleading in the network [129]. Its impact on packet loss rate and transmission delay is significantly increased by a reasonable range. Also, appropriate countermeasures for overcoming the hindrance due to Persona hijacking are elaborated in a wireless sensor networking scenario. In another work by authors in [130], by using source address validation and stateful packet supervision, the DoS attacks leading to Persona hijacking a lightweight, deployable SDN defense framework known as FloodShield was designed. It provides an effective shield against such attacks with less resource consumption in data and control planes.

6) REVERSE LOOP

The controller's view of the SDN network is highly exploited with the reverse loop attacks in the network. The controller can infer whether a reverse link exists considering the dynamic characteristics of the network. Within the stipulated time interval, the existing inter-switch links are removed in these attacks by reversing the links. Nisar *et al.* [35] summarised the impact of reverse loop attacks in SDN with their issues and challenges. With the analysis of six different vulnerabilities, the exploitation of the reverse loop introduces attacks during topology discovery mechanisms in SDN [131].

7) TOPOLOGY FREEZING

This kind of attack also affects the controller's perspective visualization of the network. It exploits the weakness of the topology services providing modules in the controllers. This kind of attack freezes the topology view of the controller and resists it from updating the dynamic changes in the network. A more detailed analysis on the impact of topology freezing is summarised in [35]. During the practical countermeasures applied to the SDN topology discovery, the impact of topology freezing resists updates of the network changes [131]. Furthermore, in another study, the vulnerabilities exposed on the control plane of SDN and its data dependencies are explored [132].

8) NETWORK MANIPULATION

Typically, network manipulation attacks against SDN occur in the control plane. In the network manipulation attack, the adversary introduces fake data in the network by compromising the controller and trying to introduce other attacks to the whole network [133]. Ubale and Jain [134] summarized the impact of DDoS attacks in SDN driven through the process of fake data injection in the network. Such exploitation of fake data predominantly affects the quality of decisions made by the models deployed for SDN-native big data applications [135]. In the work reported by Sallam *et al.* [9], the authors presented a Software-Defined Perimeter (SDP)

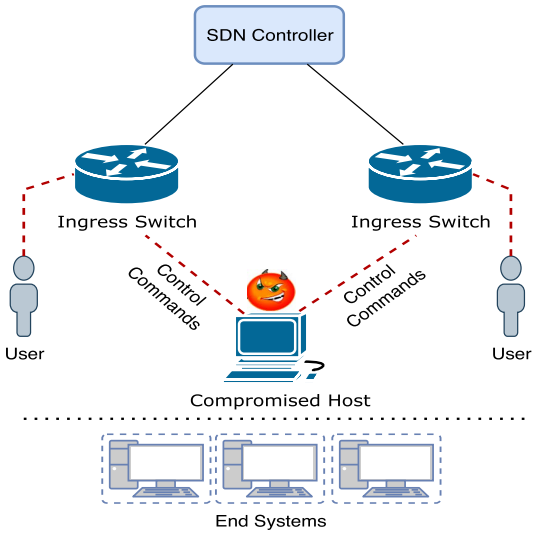


FIGURE 5. MITM attack on the controller.

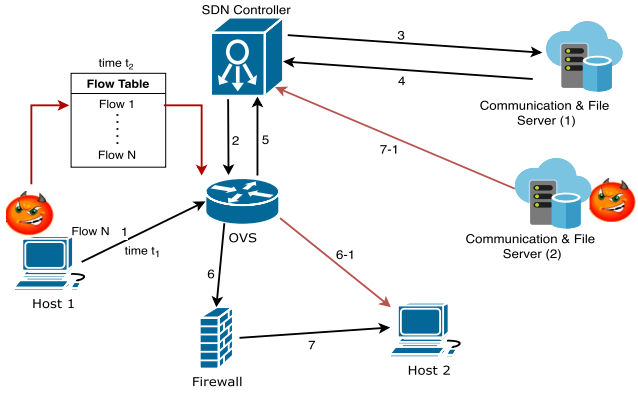


FIGURE 6. A threatening application inserts a flow rule in order to let adversary packets avoid/bypass a security application implemented in the controller.

framework and examined the impact of network manipulation attacks through virtualized network testbeds. The testing results of this framework have proven the framework to provide a flexible shield against DoS bandwidth and port scanning attacks with the established orchestration of connections. Xiao *et al.* [136] reported data dependency creation and chaining attacks due to the network manipulation in SDN. Here, using the SVHunter tool, the vulnerabilities of the attacks are effectively identified. Moreover, it could also report remote network exploitation through arbitrary commands, exfiltration, and crashing of SDN services.

9) TRAFFIC SNIFFING

Traffic sniffing in SDN permits the attacker to eavesdrop on the data in the network and snoop on vital information by capturing and analyzing the communication interface. In the case of consistent traffic scenarios of the network, they are likely to be exploited if the traffic sniffing attack is successful [9]. In an SDN environment, an attacker can exploit data that are not encrypted to block traffic from and to the controller. Using robust encryption schemes and strong passwords could suppress the impact of traffic sniffing in SDN [137].

10) MAN-IN-THE-MIDDLE (MITM) ATTACKS

Benton *et al.* [138] and Brooks *et al.* [139] present the way MITM attacks can be very severe in OpenFlow environments compared to the legacy networking environment because of the lack of authentication enforcement over the plaintext OpenFlow TCP control layer. In an SDN environment, the adversary can potentially take advantage of downstream OVS (Open vSwitch) switching devices to establish advanced eavesdropping attacks, as shown in Figure 5.

Furthermore, a fraudulent inter-link injection might trouble and impact the Shortest Path Selection (SPS) service process. An attacker has the ability to establish LLDP relay-based channels to mislead an SDN device with the illusion of a non-existing link (i.e., internal link) that connects the potentially targeted OpenFlow switching device [140]. Once the SDN device observes a link-up, it will immediately search for the shortest path utilizing compromised and infectious networking topology information. Thus, all the networking traffic crossing the fake computed path will go to the attacker's trap.

B. DATA PLANE

1) FRAUDULENT FLOW RULES

SDN controller is responsible for a typical flow rules-driven networking environment. Therefore, it must instruct and command the entire networking traffic behaviors, including the configuration of OpenFlow switching devices with flow rules, to provide consistent and valid flow rules. For this reason, the SDN controller imposes valid, secure, and dynamic mechanisms to better authenticate all flow rules between the application layer and control layer of SDN [141]. Figure 6 depicts an attack scenario where the adversary deploys an application on server B to inject/insert a flow rule in the SDN device, which then permits the OVS device to directly forward the adversary traffic to Host C. Therefore, the security application in the SDN controller is bypassed and thus rendered worthless.

2) FLOODING ATTACKS

The rate of DDoS and DoS attacks occurrence in environments of SDN-based communication systems has tremendously risen because of the on-demand self-service and various accesses to network services [142], [143]. When a network packet mismatches the flow rules of OpenFlow, it is still transmitted to the SDN controller by the OpenFlow switch as depicted by Figure 7. Regardless of decoupling data from the control plane, protocol rules are still insecure. They enable adversaries to tamper with data plane forwarding and network topology that are critical and sensitive to the appropriate operation of SDN environments. Particularly, malignant devices can form and send a fake packet to be relayed

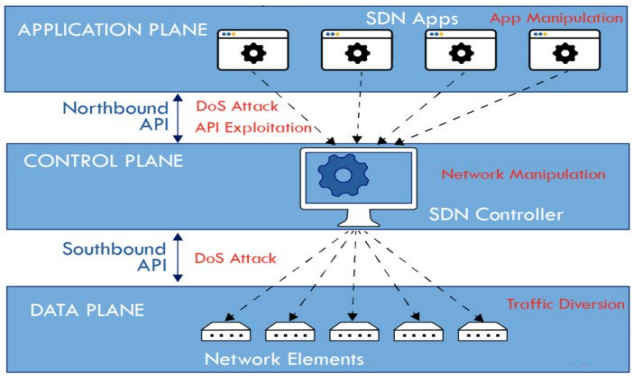


FIGURE 7. DoS attack scenarios against SDN.

by an SDN switch as a genuine data packet (i.e., packet-in) and forward it to the SDN controller, launch a DoS/DDoS attack on the SDN controller and SDN switches, or even elicit useful information of flow rules based on side-channel mechanisms.

Furthermore, to block the broadcast storm and economize energy, the SDN controller fits Spanning Tree Service (STS). Once an update in the topology takes place, STS prevents redundant ports. Anywise, this ability could be leveraged by an attacker to trigger a DoS raid. Also, an attacker can disable the OpenFlow switch ports by grouting fake links into SDN topology. Thus, DoS/DDoS is one of the most harmful and menacing attacks on the SDN controller, as Shin *et al.* [144] have demonstrated through a proposed scanning tool. The developed tool is capable of characterizing DoS/DDoS threats based on the response time of network flows.

- SYN flooding attacks: On the one hand, since an SDN controller permits for routing of the network flows via a centralized firewall in order to protect and secure the SDN data layer, this type of operation may lead to exhausting the resources of the OpenFlow switching devices (through fraudulent flow rules) as handling the communication traffic between the SDN controller and OVS devices may take significantly a long time [145]. On the other hand, installing (i.e., insert/push) flow rules by an SDN controller may be challenging with regard to traffic flow inspection, authorization, and authentication throughout security software and implementations.
- TCP-based flooding attacks: These typical attacks can be established by transmitting a large amount of TCP-based traffic with only the SYN flags set, which leads to clogging the network bandwidth. Additionally, the fact that OpenFlow switching devices need to buffer traffic flows while the SDN controller is issuing the flow rules renders the SDN data plane vulnerable to TCP-based saturation attacks. The OVS possesses constrained resources to buffer stop the unsolicited TCP and UDP-based traffic.

3) TRAFFIC DIVERSION

In the data plane of the SDN framework, the network elements are prone to traffic diversion issues that can be introduced by adversaries. It leads to eavesdropping by exploiting the network elements and redirecting the network traffic flow. A Dynamic Traffic Diversion (DTD) [146] algorithm was utilized to test traffic diversion issues in network emulation using Mininet and Cisco equipment. It was observed from the experimental results of test cases that the dynamic diversion of traffic helps to prevent the jitters and reduces the packet losses in the network. Further, the authors in [147] proposed a support vector machine (SVM)-based Internet traffic identification and classification mechanism that could be effectively employed for identifying the application traffic in the network. Here, the experimentation is carried out on social media traffic, and the classification accuracy of application traffic in different social media platforms was presented and analyzed.

4) ARP SPOOFING ATTACK

Address Resolution Protocols (ARP) are meant for identifying the MAC address associated with the IP address. ARP spoofing attacks take advantage of this process and enable the attacker to transmit fake ARP messages to the LAN in an attempt to associate their MAC address with the IP address of different hosts [148]. This attack is also limited to local network segments, and it can facilitate MITM attacks. In this process, two SDN hosts think they are communicating with each other, but they pass through a third party network module [149]. The authors in [150] proposed three different strategies for the detection of ARP spoofing attacks. This includes signature-based, ML-based, and Wireshark-based packet analysis techniques. Further, an accuracy analysis was elaborated based on the strategies for detecting ARP spoofing attacks, better protection against spoofing was observed.

5) SIDE CHANNEL ATTACKS

The weaknesses associated with cryptographic algorithms that lack strong mathematical aspects can be exploited by side channel attacks during their implementation. Moreover, these kinds of attacks are mostly non-invasive and passive (i.e., they will not leave any traces of the attack). They normally use the leaked signals from side channels during the normal operations of the channels for revealing the data in SDN. Side channel attacks typically make use of the delay encountered in the network to guess the network configurations. Introduction of time our proxies could reduce the response time for countering the side channel attacks [151]. Also, dynamic adjustments of workloads in the control plane are required to cope with such attacks.

Authors in [152] developed an elliptic curve Diffie-Hellman (ECDH) based key exchange mechanism and incorporated strong countermeasures to thwart the effects of side channel attacks. It was experimented on low footprint

embedded devices using Four$\mathbb{Q}$, which ensures providing rich arithmetic operations and a secure implementation for the low power devices.

C. APPLICATION PLANE

1) APP MANIPULATION

In the application plane of SDN, the applications are prone to App manipulation attacks. In these attacks, the attacker may gain access to an SDN application, cause malfunction, eavesdrop on the data involved, and disrupt the services. Also, the attacker could gain higher privileges to perform illegal operations on the SDN applications.

2) API EXPLOITATION

Application Programming Interface (API) of specific software components linked with SDN systems are prone to be exploited by attackers, leading to unapproved exposure of vital data. This ends up with API exploitation, which may lead to the destruction of the flow of information in the network. Proper updates on the patches of applications running on SDN nodes are mandatory to avoid such exploitation.

3) PASSWORD GUESSING AND BRUTE FORCE

Through a brute force attack, the hackers attempt to find the user credentials by trying out various possible credentials through all possible combinations and permutations of passwords and usernames. This attack guesses the password through the trial and error method.

4) AUTHENTICATION AND AUTHORIZATION

Kreutz *et al.* [153] introduced vulnerability vectors in the SDN environment. It is demonstrated that the SDN controller does not impose any compulsory enforcement for instituting a trusted relationship among applications (i.e., SDN applications, networking service applications, security applications, etc.) and the SDN controller. As a result, malignant user applications could establish severe damage to the entire SDN-enabled environment as the SDN controller allows for layers abstraction, which is explicitly interpreted into configuration commands of the SDN infrastructure by different applications.

Furthermore, the credentials of users in a particular application could be compromised if an application server containing the specifics of these users is compromised. These compromised credentials might lead adversaries to insert counterfeit traffic that is regarded as authorized into the network. To defend against such threats, a centralized security method needs to be installed in SDN. However, there is no such mechanism developed yet. In OpenFlow specifications, the SDN applications make use of a broad range of features, characteristics, and networking services provided by the SDN controller. These SDN applications could be implemented by any third party rather than the SDN controller vendors only. Therefore, the access privileges and authentication mechanisms should be imposed in such a programmable SDN environment as described in [154]. However, it is very challenging to accomplish so due to the remarkable growth of SDN and security applications.

5) ACCESS CONTROL AND ACCOUNTABILITY

As an SDN controller allows for implementing/installing a broad range of applications to better utilize the network and SDN services, these applications are granted quietly powerful access authorities, rendering the entire SDN environment insecure [155]. For this reason, constrained access control and authority enforcement mechanisms need to be imposed on the implemented applications in order to guarantee the security of the networking environment. To better comprehend the security vulnerabilities associated with authority and access control of SDN applications, Figure 8 depicts some security weaknesses in the application plane of the SDN controller. The applications impacting the SDN environment security are classified into three main categories based on [155]:

- Applications that are networking-sensitive, demanding specific network advantages/features (e.g., cost of the path.)
- Applications that deliver network servings (e.g., IDS/IDPS services, packet/flow content inspection, etc.)
- Tinned applications that integrate/merge other applications from categories one and two (i.e., giant applications that require instantiating a particular virtual component in the networking environment.)

Threatening security applications can lead to network traffic skipping authority and access control imposed by the SDN via the use of one or multiple of the aforementioned category application. Thus, a compromised SDN application might establish a gateway for illegitimate and unauthorized applications to access the control layer of SDN.

6) UNAUTHORIZED ACCESS AND SCALABILITY THREATS

As the SDN creators aimed to attain nimbleness and scalability from the network management perspective, SDN components gradually utilize an elastic cloud infrastructure and a flexible allocation of resources. However, the SDN and networking applications built in the control layer could cause dangerous vulnerabilities to this layer [155]. Furthermore, the authentication and authorization capabilities are a remarkable challenge in SDN environments due to the weaknesses posed by the controller when it comes to separating varied security applications of various implementations' implications:

- Different applications have different functional requirements from the underlying SDN, and data paths should qualify for various security requirements. (1) Applications may need network statistics (e.g., bytes or packets counter information) from the switching device to proceed with load balancing processing [156]. (2) IDS/IDPS implementations could require an inspection of packet header information.
- The need to authenticate and authorize third parties which may apply an auditing operation for each implemented application.

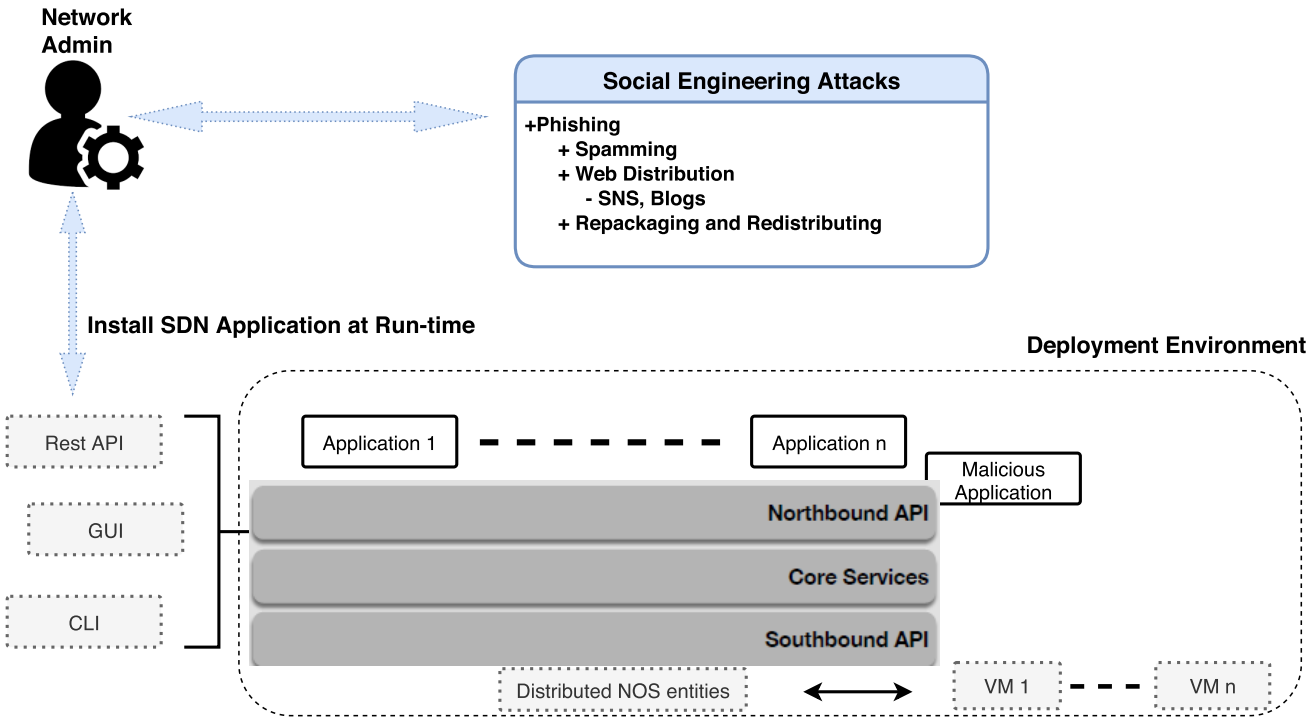


FIGURE 8. SDN grants applications powerful authority and access which could be very threatening.

- Authenticating and authorizing resources deployed by various applications with an appropriate tracking and isolation.
- Network providers and networking operators need to possess a variety of higher-level privileges in order to gain access to networking resources. Typical user applications need to acquire a partial role in the network configurations and management [157]. Therefore, these applications need to be appropriately inspected prior to granting end-users privileged access to resources in the network.

Based on the challenges mentioned above in the control layer with regard to control applications, a careful and customized enforcement of security policies is needed in different sorts and levels of implemented applications (i.e., could be achieved via SDN northbound API). However, this has not been expounded and addressed yet.

Moreover, from the perspective of controller availability, when the amount of traffic flows considerably increases in a networking environment managed by only one SDN controller, the overall processing time may become significantly impacted, leading to a typical point of failure. Yao *et al.* [158] have also tested multi-domain SDN controllers and multi-SDN controllers in a single domain and proven this typical point of failure is feasible. Furthermore, scalability in SDN controller renders DDoS/DoS attacks even more feasible because an SDN controller in a high-speed environment can quickly get a bottleneck while inserting rules of network flows into OpenFlow switching devices through the data-path [159]. This scalability challenge might plausibly lead to control layer saturation threats [160].

D. CROSS-LAYER ATTACKS

1) CrossPath ATTACK

The control channels of the SDN are prone to CrossPath attacks, in which the attacker exploits the shared paths among the data and control signal traffic. The attacker may introduce hand-crafted data for imparting disruption in the control traffic, particularly among the shared links of the network. However, the attack is highly sneaky, and the controller could not easily perceive them, as the data traffic is not effectively combined with the control channel signals. Cao *et al.* [161] experimented using an SDN testbed to study the impact of the CrossPath attacks on network applications. Its impact primarily leads to network-wide DoS, routing blackhole, and flow table resetting. These anomalies significantly degrade the performance of the network applications. In another work by authors in [136], a novel tool named SVHunter is developed to evaluate the unexpected data chaining and data dependency created due to the Crosspath attacks. A Fast recovery Saturation attack Detection and Mitigation framework (FSDM) [162] enables a novel functional module to help the network to recover from the attacks much more quickly.

2) TELEPORTATION ATTACKS

As discussed earlier, the decoupling of the control-data plane in SDN grants a reliable programmable network. However,

such a promising paradigm enables various teleportation vulnerabilities (switch and host-based attacks) in SDN environments [163]. These vulnerabilities are summarized as follows.

- Network functions bypassing and policy conflicts evading: Exploitation of teleportation to avoid middleboxes that enforce security checks, billing, or QoS policies.
- Rendezvous and malicious switch discovery: Malignant or Trojan switching devices such as OpenFlow switches that hold a hardware/software backdoor can exploit teleportation as a rendezvous protocol to identify each other and coordinate a security attack.
- Exfiltration: Exploitation of teleportation to exfiltrate critical data within networks without data layer connectivity.

3) ROOTKITS THREATS

The application-controller interaction is one of the advantages that the SDN architecture offers to ensure the network operability [164] in the context of reliability and networking automation. However, recent studies [164]–[166] have demonstrated that existing solutions for security and policy enforcement in SDN control lack efficient security properties. Particularly, SDN rootkits can permit adversaries to control an entire SDN-enabled environment by compromising one or multiple controllers [164]. This is feasible due to the lack of network policies enforcement for application implementations to detect and prevent malignant network programming attempts [166].

4) CONTROLLER PLACEMENT THREATS

In an SDN environment, a controller is in charge of enforcing security policies. Thus, incompatible or inappropriate configurations of multiple controllers in both single-domain and multi-domain environments may cause internal conflicts (e.g., there is no guarantee that all deployed SDN controllers will obtain information about network state and resources once a network change or update takes place [167].

Moreover, when more than one SDN controller is deployed in a networking environment, this splits the entire networked system among the multiple controllers, consequently creating sub networking domains. Therefore, preserving and attaining security policies and applications in each individual SDN domain could be challenging or even infeasible.

E. LESSONS LEARNED

Table 4 gives a classification of key security attacks in SDN infrastructure. The attacks are classified per SDN plane. The worth of SDN lies in its capability to guarantee coherent policy enforcement and better scalability due to its centralized management and network programmability [31], [168]. The future generation of security solutions will benefit from the richness of network usage information available in SDN to enhance security policies enforcement, network anomaly revelation, and attenuation [8].

For simplicity reasons, in our paper, we discuss security threats associated with each of the three SDN layers, control layer, application layer, and infrastructure layer, separately. However, as SDN technology is widely and gradually growing, the catalog of security vulnerabilities is anticipated to evolve in the near future vastly. Based on Figure 9, a summary of security threats and challenges related to the three planes of SDN is presented as follows.

1) CONTROL LAYER

- Unauthorized controller access: No compelling mechanisms for enforcing access control on applications.
- Availability and scalability: Centralizing intelligence in one entity will most likely have scalability and availability challenges.
- DoS/DDoS attack: Visible nature, centralized intelligence, and limited resources of the control plane are the main reasons for attracting DoS/DDoS attacks.
- Distributed control layer: When multiple SDNs are deployed into control a tremendous number of devices (i.e., through splitting the network to various sub-domains), the ingathering of information and aggregation of flow rules within each sub-domain is a challenge.

2) INFRASTRUCTURE LAYER

- TCP-based attack: Transport Layer Security (TLS) is susceptible to TCP-level attacks.
- Flood-based attack: Flow tables of OpenFlow switching devices could hoard/save a fixed or limited amount of flow rules.
- Fake traffic/flow rules: Data plane is dumb and therefore more susceptible to fraudulent flow rules.
- Hijack of the controller: The infrastructure layer in SDN is solely dependent on the control plane, making its reliability dependent on controller security [169].
- MITM attack: Due to optional use of TLS and complexity of configuration in TLS.

3) APPLICATION LAYER

- Accountability and access control vulnerabilities: It is challenging to carry out accountability and access control policies on third-party applications and nested applications that consume network resources.
- Authentication vulnerabilities: No compelling authentication and authorization mechanisms for applications and more threatening in case of a large number of third-party applications.
- Fake insertion of traffic/flow rules: Malignant or compromised applications could create false flow rules, and it is troublesome and challenging to verify whether a particular application is compromised.

IV. SECURITY FRAMEWORKS AND IMPLEMENTATIONS

In this section, we first overview the security frameworks behind developing SDN solutions and address the challenges associated with their implementations. Then, we introduce

TABLE 4. Classification of security attacks per SDN layer.

SDN Layer	Attack Type	Attack Goal & Impact
Application	Unrestricted authority	Application termination
	Malicious apps	System commands execution
	Service neutralization	Manipulate control packet handlers. Execute a service disruption. Sniff sensitive network information. Execute specific deviant actions
	Vulnerable northbound APIs	Terminate a victim application. Issue a system command. Expose information exchanged between controller and a target application
Control	Flow rule tunneling	Circumvent firewall. Instruct conflicting flow rules.
	Controller poisoning	Poison controller information and topology view. Propitiate execution of attacks on data plane. Poison network topology with crafted LLDP packets
	Host Location Hijacking	Insert fake network entries. Poison controller host profile reservoir
	NOS misuse	Exploit controller misconfigurations. Force controller termination. Redirect sensitive information. Install rootkits. Insert invalid input data leaving the controller in unpredictable state.
	Packet-in flooding	Broadcast massive malformed packets. Increase switch table misses. Overwhelm controller responding to packet-in messages via south-bound API
	Controller switch table flood	Continuous reception of forged features-reply packets. Store fake switches entries in the controller switch table. Degrade the controller performance
	Forced switch disconnection	Legitimate switch identity hijacking by impersonating a trusted switch DPID. Poison the Spanning Tree. Deploy a malware in NOS to corrupt the controller switch table entries
	Channel eavesdropping	Leverage unencrypted control channels. Perform packet sniffing. Listen to control, topology, and management traffic
	MITM	Infiltrate communications between controller and the target data plane's device
Data	DoS/DDoS	Exhaust switch/controller resource. Isolate target switch
	ARP poisoning	Hijack controller identity. Force connection drop by the target switch. Force switch to connect to a fake controller
	Flow rule manipulation	Modify entries in switch flow-table. Overwrite existing flow-rules via a compromised controller application
	Flow rule flooding	Forcing a switch to raise a fake flow rule request to the controller. Force a switch to continuously request new flow rules to flood its flow table. Degrade the switch stability and response time
	Control packet injection	Expose a switch to a fuzzing-attack situation. Inject crafted control packets containing malformed headers into the switch table
	Side-channel attacks	Infer about specific information to perform another type of attacks (e.g., perform a flow table flooding on a switch by inferring about RTT)

the security enhancement strategies at each layer of SDN. In particular, we discuss how to impart robust security features with the aid of other supporting technologies. In SDN architecture, the control layer is logically detached from the data plane to elaborate decisions in a decentralized manner due to a holistic view of the global network environment. Moreover, the centralized architecture of the SDN controller grants a reactive control and analysis of security and alteration of security policies [170], and therefore facilitates efficacious alteration of security policies to implement/insert security policies and drive them into the network entities and components. This minimizes the risk of improper policies configurations and policy conflicts over the SDN environment.

Despite the fact that SDN architecture preserves a grand promise to flexibly facilitate network management and deployment about lowering the overall cost of network management, numerous security challenges need to be addressed. This section presents and discusses common security measures, existing research proposal advances, and solutions dedicated to resolving the security challenges in the control, applications, and data layers of SDN.

A. CONTROL PLANE SECURITY ENHANCEMENT

The control layer is in charge of granting unified and consistent management of the OpenFlow-enabled forwarding devices and higher-level applications that carry out networking features-based service calls (e.g., link discovery service, entry tables, etc.). Therefore, the SDN control layer needs to be secure against the following key threats:

- DDoS/DoS threats
- Mal-replacement of SDN controller, which threatens the availability
- Anomalous applications
- The fudging activities that threaten the scalability of the control layer

As the SDN controller grants applications access to different networking inputs and resources via its control layer, it becomes of a great substantiality to protect the control layer from potential anomalous applications and enforce controlled access to the SDN-enabled applications. Table 5 shows existing solutions (e.g., proposals, platforms, and frameworks) proposed to address security threats and challenges in the SDN control layer.

SDx [171] provides a Floodlight-based SDN controller [42] extension. The designed framework [171] is a security-enhanced (SE) version of the control layer in Floodlight controller [42] that offers various methods for prerogative decoupling through a flexibly programmable northbound interface. Additionally, the mechanism allows the SDN controller to behave as a medium between SDN applications and the SDN infrastructure layer using checking techniques for legitimizing the class module responsible for managing networking flows.

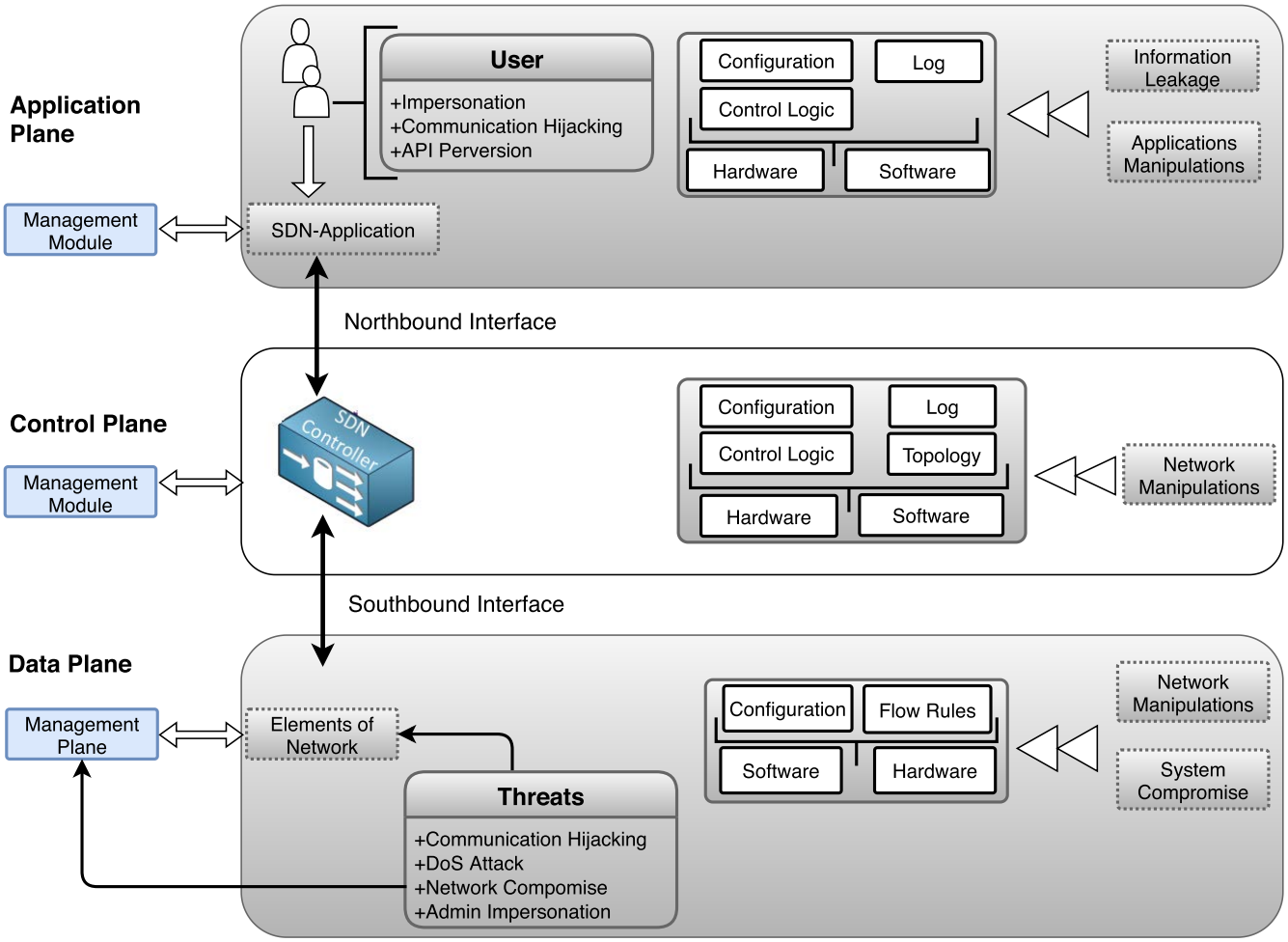


FIGURE 9. An SDN architecture comprising the three planes with associated threats.

According to OpenFlow specifications, the SDN controller must insert various flow rules for all customer entities' connections. As a result, this causes a bushy load in the SDN due to the massive amount of flow rules that the controller needs to install in the switching devices. For this reason, several proposals and mechanisms were presented to reduce the SDN load and/or divide it between the control layer elements and functionalities.

In [172], Fernandez presented comparisons between pro-active and re-active models of SDN scalability. While a pro-active SDN installs the networking flow rules before the traffic arrival in the Open vSwitch (OVS) using specific predefined transmission rules, a re-active SDN first extradites the packet and then inserts the flow rule in the OVS for this particular packet. Fernandez [172] also showed that the re-active SDN is significantly less scalable in contrast to the pro-active one. Because the pro-active SDN must be informed about all networking flows beforehand, which is infeasible, Fernandez [172] proposed a hybrid SDN that behaves interactively to set the paths while comprehending the flow conduct to determine the route beforehand proactively.

In [48], Tootoonchian *et al.* proposed HyperFlow, as an event-based distributed but logically centralized controller framework. The proposed platform warrants network administrators to increase the SDN scalability while decreasing the time needed to configure networking flows. In [175], Heller *et al.* proposed SDN placement in a way that lowers the flows latency through the use of a load-balancing approach to distributing the load on multiple SDNs. However, the proposed approach stipulates a trade-off between the state allocation and SDN elements availability. In [173] and [174], a physically distributed SDN control layer named DISCO is introduced to transit the operations of the SDN control layer in distributed and overlay networks. The proposed framework can be configured on top of Floodlight [42] SDN controller and deploys AMPQ [190] specifications.

Furthermore, there have been several proposals in the past addressing the SDN processing power and engaging the charge split between multiple SDNs, such as [191], which is an extended version of SDN with multiple CPUs designated to boost control algorithms at-scale. The presented framework demands a holistic view of networking state alteration

TABLE 5. Implementations and solutions for control layer security enhancement.

Security Solution	Targeted Threat	Solution Type
SE-Floodlight [171]	Applications authorization	Framework for security applications development
HybridCtrl [172]	Control scalability	Applications control system
DISCO [173], [174]	Control scalability	Applications debugging framework
Ctrl-Placement [175]–[177]	Control availability	Security policy verification applications
HyperFlow [48]	Control availability	Applications testing framework
DDoSDetection [178]	DDoS attack	Detection framework
FleXam [179]	Malicious packet and middleboxe inspection	Sampling action framework
CloudWatcher [180]	Network flow checking	SDN monitoring application
L-IDS [181]	Malicious configuration of embedded OpenFlow devices	IDS
Avant-Guard [160]	Data-to-control bottleneck and DoS	Detection and mitigation framework
HSA [182]	Failures (reachability, forwarding Loops, and leakage threats) caused by malicious configurations	Network policy checking
NetPlumber [183]	Applications Compliance of network state changes	Real-time policy checking
PermOF [154]	Application authorization	Fine-grained permission system
Betge et al. [184]	Path exploitation	Trusted path enforcement in a multi-domain SDN environment
Cox et al. [185]	Network policy violation	Security policy transition via the controller
SDN-Guard [164]	Malicious network programming	Dual view-based framework for preventing SDN rootkits
Melis et al. [186]	Network policies violation	ONOS toolkit prototype based on four application plug-ins
Varadharajan et al. [187]	Service, host, and user policy violation	Secure architecture for policy enforcement
SMCDS [188]	Reconnaissance attacks	Distributed shadow controller architecture
POISE [189]	network policies	Programmable in-network security for context-aware policies

incurring at the incoming traffic rate. Additionally, the framework is much more scalable than its counterpart NOX and it could be extended through high-level programming and scripting languages.

Besides, the DoS/DDoS threats could be practically alleviated through the network conduct dissection or traffic statistics saved in OVS devices. Although the network behavior characteristics are facilely purported and extracted from OVS switching devices, fetching them might be costly because of the low overhead support in these OpenFlow switching devices. In [178], Braga *et al.* proposed a lightweight mechanism to detect DDoS threats based on self-organizing maps (SOM) [192] to identify the concealed associations and connections between network flows crossing the environment. A Self-Organizing Map is a neural network for an n-dimensional pattern of data transformation into one or two-based dimensional maps. Here, the procedure of data transformation provides a gathering of data patterns that has identical statistical behavior for additional processing.

As load-balancing is another vital method to mitigate DDoS attacks, Belyaev *et al.* [193] proposed an approach to mitigate DDoS threats by considering the load-balancing problem in the SDN controller. In [178], the proposed technique for DDoS attacks capture employs three main components; classifier, which classifies the data extracted by other components according to the corresponding flow (malicious or benign flow) using SOM [192], feature extractor that collects statistics and characteristics that are vital for DDoS occurrence, and the flow collector module that collects traffic entries from the OVS flow tables. The collected characteristics include, but are not limited to, the average time interval and the number of packets per incoming flow.

SPHINX, which is presented in [194], aims to detect attacks that contravene the learning-based flow modules by designing a network flow graphs-based prototype. TopoGuard [140] is an extension to the security side on SDN controllers, which captures attacks that attempt to poison SDN environments (i.e., holistic visibility of network environment/topology) based on security omission's fixation. However, TopoGuard's prototype evaluation was based on the Mininet emulator, not on a real networking testbed. RAID [195] is another control prototype to passively monitor the network systems and target operational exploitation in a large-scale environment. However, the effectiveness of the prototype was assessed only through OpenFlow-backed connecting to three hardware switches.

Furthermore, several recent advances in SDN security have primarily targeted only one type of threat and proposed a security enforcement framework to mitigate it. Significant studies have extended the controller software for security enhancement against DoS threats such as [196], [197], and [198]. Most notably, SDNScanner [144] and AVANT-GUARD [160] provide a solution to mitigate and solve the problem of saturation attacks (data-to-control layer saturation) by altering flow management at the switch level. However, the presented design [160] is limited to TCP saturation attacks and exposes only the flows that complete the TCP handshake. Since this design is based on an SYN proxy implementation, it is unsuitable for different protocols.

VeriFlow [199] detaches a holistic network environment into sub-classes that have exactly similar forwarding

behaviors exploiting a multi-dimensional prefix tree so that all forwarding policies will be checked in a live time whenever a network update occurs. NetPlumber [183] is a real-time policy verification tool based on Header Space Analysis (HSA). VeriFlow [199] and NetPlumber [183] assume the data plane is threat-free while checking the policies' constraints, and therefore do not consider investigating a broad class of inter OpenFlow switch-controller packets to refine the behavior and performance of the networking environment. In [200], Chung *et al.* introduced NICE, an approach to detecting network software bugs in OpenFlow-enabled applications based on symbolic execution and model checking.

In [175], Heller *et al.* discuss the SDN controller placement challenge. At the same time, they demonstrate the quantity of SDNs is one of the main defiances to establishing a resilient and scalable SDN environment. Thus, the controller placement problem has gained remarkable attention from the networking industry and researchers, and various mechanisms have been suggested and evaluated, such as [176], [177]. Bari *et al.* [201] proposed probabilistic-based Simulated Annealing (SA) algorithms to solve the controller placement problem optimally. In [176], Hu *et al.* portrayed the problem of controller placement as an NP-hardness problem as one way to enhance the SDN reliability in terms of functionality. In contrast, the quality of time responsiveness is preserved. Although the presented solution enhances the network behavior and operation reliability by reducing the anticipated rate of control-path loss, several recent algorithms for solving the same problem are eventually evaluated in real networking topologies, and trade-offs between reliability and response time are discussed.

Additionally, the dynamic controller provisioning problem (DCPP) was targeted by Bari *et al.* [201]. A platform is presented in this work for a real-time adoption of multi-SDN controllers in a WAN. Here, the locations and quantity of deployed controllers are regulated based on network mobility. Besides formulating the DCPP as a linear problem, which utilizes the networking flow paradigm and characteristics to reduce the OVS state collection expense (i.e., communication overhead), the authors demonstrated trade-offs between communication cost and the time required for flow configurations. In [202], Hock *et al.* introduced a lightweight mechanism for SDN placement based on Pareto to improve the controller availability. Based on [202], while a single SDN can be sufficient to fulfill the latency requirements, additional SDN devices must be deployed to satisfy the network resiliency specifications.

The OpenFlow SDN procures the topology state datums through LLDP, a link layer protocol to collect information about OVS devices connected to the SDN controller (e.g., the status of connection, switch information, connection port information, etc.) [203]. To solve the problem of SDN involvement in all LLDP packets that might lead to scalability challenges in SDN controller, Kempf *et al.* [204] present a practical design that offloads the potency of topology monitoring from the SDN to the OVS. Although the proposed platform provides a generic message simulator for OVS devices, its monitoring features and modules are compatible only with OpenFlow protocol version 1.1. In [205], Monsanto *et al.* introduce NetCore, a high-level language to describe flow rules and policies in the OpenFlow controller using a dynamic compilation-based mechanism to split the charge of flow handling between the SDN and OVS devices.

In [206], Mogul *et al.* benefited from the intrinsic trade-off between the OVS devices and the controller to enhance the availability of SDN infrastructure to design DevoFlow. The DevoFlow mechanism adjusts the OpenFlow architecture to reduce the cooperative computations and communications between the infrastructure and control layers. Because the holistic visibility of all network traffic is not indispensable, Mogul *et al.* designed the DevoFlow to incline certain control operations back to the OVS device while only preserving the centralized control operations in the SDN controller.

In [164], Tatang *et al.* presented SDN-Guard, a mechanism to detect and prevent SDN rootkits that permit adversaries to control the network in an SDN-enabled environment via compromising a controller and thus, subverting the network operating systems [166]. SDN-Guard [164] performs a dual-view comparison to identify malignant network programming attempts and hence block them. Moreover, Melis *et al.* [186] gave an SDN-enabled toolkit to identify and prevent various threats aiming to compromise and exploit the forwarding operations. The addressed threats include bogus injected flow rules-based compromised network boxes, inner loops, and black holes that are uneasy to identify using normal network scans.

Furthermore, Wang *et al.* presented SECOD [207], [208], an SDN secure control and data plane algorithm, which deploys new triggers to identify and prevent DoS attacks in both control and data layers of SDN. SECOD [207] is implemented on SDN-enabled hardware switches. In [187], Varadharajan *et al.* introduced a security architecture for policy enforcement in SDN. The architecture is based on a language technique to design security policies for protecting SDN services and control flows in a multi-domain SDN. Last, In [130], FloodShield is presented by Zhang *et al.* to solve the DoS-enabled data-to-control saturation attacks. FloodShield [130] is a deployable and lightweight IDPS solution to thwart dedicated saturation attacks through leveraging source address validation and stateful packet supervision according to evaluation scores and resource utilization.

B. DATA PLANE SECURITY ENHANCEMENT

The SDN data/forwarding layer is eventually in charge of assuring and guaranteeing the appropriate transit of network flows from an input port (i.e., ingress point) to a destined output port (i.e., egress point). Specifically, this transit follows the *match* : *action* rules inhibited in networking-enabled entity forwarding tables or FIB. The forwarding layer needs to be protected from anomalous applications capable of inserting new flow rules or even adjusting existing rules within the forwarding layer. Therefore, earnest enforcement

TABLE 6. Implementations and solutions for data layer security enhancement.

Solution	Targeted Threat	Solution Type
FortNOX [209]	Flow rules contradiction	Controller framework
FlowChecker [167]	Faulty flow rules	Configuration verification tool
VeriFlow [199]	Faulty flow rules	Network debugging tool
Resonance [210]	Access control	Access control and policy enforcement framework
CPRecovery [211]	Controller availability	Controller replication framework
FleXam [179]	Malicious packet and middleboxe inspection	Sampling action framework
L-IDS [181]	Malicious configuration of embedded OpenFlow devices	IDS
Avant-Guard [160]	Data-to-control bottleneck and DoS	Detection and mitigation framework
HSA [182]	Failures (reachability, forwarding Loops, and leakage threats) caused by malicious configurations	Network policy checking
ClickOS [212]	Data plane programmability	A Xen-based virtual machine to run a wide range of middleboxe
Nanda et al. [213]	IP-based malicious logins	A ML-based solution for malicious traffic patter prediction
Honeyproxy [214]	Data-control malicious flows	Design and implementation of next-generation SDN-enabled Honeynet
SHIELD [215]	DDoS	NFV/SDN-based IDPS
Zhou et al. [216]	Information leakage	Transmission channel-based security service via multi-connection and time-slot scheduling
OFMTL-SEC [217]	Exploitation of network function implementations	Stateful date plane solution
DPX [218]	Complexity of security services configuration	Abstraction-based architecture for security services instantiation
PivotWall [219]	Malicious data flows	SDN-based flow control and tracking
SYNGuard [61]	SYN flood	Dynamic threshold-based attack detection and mitigation

solutions of security (e.g., authentication mechanisms) need to be implemented against these anomalous applications that tend to modify the valid flow rules. Table 6 depicts a list of existing solutions to enhance security in SDN controllers by addressing security threats residing in the data layer.

FlowChecker [167] is a verification and configuration solution that distinguishes and sets apart the malicious flow rules in an individual OVS device or internally-federated elements of the datapath. This security tool can also be utilized to check and examine the OpenFlow configurations at run-time for the OpenFlow applications and the master SDN controller. VeriFlow [199] is another mechanism for networking debugging and utilized to identify erroneous flow rules that were pushed by particular SDN-enabled applications and prohibit them from occasioning malicious network demeanors. Monsanto *et al.* [209] proposed FortNox, a framework that uses a real-time engine for flow rules examination. The proposed solution allows a NOX-based SDN controller to verify any discrepancies in networking flow rules asynchronously and dynamically at a run time. It also grants the newly implemented SDN applications the necessary authorization before modifying any flow rules in OpenFlow-enabled switching devices. Additionally, the proposed platform [209] delivers authorizations for OpenFlow applications based on the role via NOX OpenFlow controller software-based extension.

More studies have primarily concentrated on securing the infrastructure layer from malignant applications. Among these works, FortNOX [209], which addresses the SDN tunneling attack and solves the rule conflicts that contravene the security policies by designing a security enforcement kernel for SDN that aims to identify rule conflicts and resolution. Rosemary [220] is another solution that adopts a practical approach to address the issue of control layer resilience through an extension of a Network Operating System (NOS) design.

Moreover, as the SDN controller is lusty in terms of functionality of OpenFlow switching devices connectivity, the superfluous connectivities need to be consistent at a run time for the inter-communication between the SDN controller and OpenFlow-enabled switches. Further, the network access control applications demand intensive network state information in the SDN environment, which implies limitations in terms of network services support. Han *et al.* [221] presented STATEMON, an OpenFlow extension for programming the stateful SDN infrastructure layer. The proposed solution is a connection tracking based that provides a global state-awareness to enhance the access management in SDNs.

Furthermore, the OpenFlow protocol grants the easiness to set a subaltern connectivity with a backup SDN device to utilize if the primary SDN comes off. Fonseca *et al.* [211] presented a replication mechanism for the SDN controller to preserve the functionality of the switching device in case the primary SDN comes off. In this solution, the OVS will recurrently transmit probing packets to the controller. If the controller does not respond within a limited time frame, the OVS device will switch (e.g., reconnect) to the backup SDN controller based on the assumption that the primary SDN is no longer in-line.

Besides, the appropriate fragmentation and planning in a networking environment are of significant importance when enhancing the security of OVS devices, leading to the optimization of the controller and connectivity efficiency. Based on the actuality that an OVS device continuously connected to the SDN will be unlikely to experience a saturation threat

as it is not imposed to save the unsought networking flows for a prolonged time frame. In [222], Zhang *et al.* show the route extent within an SDN and OVS entity is eventually proportionate to connectivity loss. Based on this assumption, Zhang *et al.* proposed that a route extent within the SDN and OVS device need to be minimally shortened to have an efficient functionality about latency qualifications and constraints and rapid analysis of security applications.

Additionally, Shaghaghi *et al.* [223] propose WedgeTail, an IPS dedicated to improving the security in the SDN infrastructure layer. The threat detection in their proposed solution is based on prioritizing OpenFlow switching devices before inspecting the networking flows using an unsupervised trajectory-based sampling mechanism where the switches are considered points within a geometric space.

Further, in [217], Scott and Arumugam presented OFTML-SEC, a state-based security solution for preventing the exploitation of network function implementations and configuration-based attacks (e.g., topology and path update, as well as ARP) at the data plane level. OFTML-SEC [217] is placed in the SDN switch and does not require the controller intervention. Particular virtualized security services often require complex configurations, leading to bandwidth drain and packet delay constraints. To address such a challenge, Park *et al.* [218] introduced DPX, as one of the security extension architecture that uses action-based abstraction for security services and translates them to OpenFlow rule sets.

Dedicated DoS attacks aim to saturate the control-to-data channel (i.e., control-to-data saturation attack), where adversaries flood bursty traffic to trigger massive packet-in messages and table-misses in the infrastructure layer, exhausting the TCAM and buffer memory. To alleviate the control-data saturation attacks, Zhang *et al.* [130] developed FloodShield, a solution that leverages a filtering approach to filter forged packets and a monitoring technique to monitor flow states of addresses before performing dynamic countermeasures using resource utilization and evaluation scores.

C. APPLICATION PLANE SECURITY ENHANCEMENT

Since the SDN controller grants flexibility for applications, while behaving as a medium bridge between the networking hardware and SDN-enabled applications, it becomes easier to implement applications that take advantage of network traffic characteristics and statistics to carry out sophisticated security features and services. Numerous languages have been introduced to design and integrate new security applications in SDN controllers, such as NetCore [205] and FRESCO [224] that allows the development of security applications complying with OpenFlow standards specifications. Table 7 presents existing platforms/solutions to enforce different types of security issues in the SDN application layer and to assure applications' compliance with OpenFlow standardization. In this subsection, we present a variety of security solutions and proposals that were introduced to verify the compliance of SDN-enabled applications with the security policies specifications.

The application layer provides various applications and networking-enabled services, including, but not limited to, Deep Packet Inspector (DPI), IDS/IDPS, and security monitoring services. Therefore, a proper level of authorization and controlled access should be enforced for these SDN applications. Guaranteeing commands and inquiries from an application that is not modified or rigged is necessary for SDN environments. However, steadying and maintaining a reliable trust between the control layer and application layer is a substantial challenge in SDN security. For this purpose, the OpenFlow-enabled devices need to be permitted to communicate and collaborate with the SDN controller at run time but without enduring a mistrusting relationship.

In [154], Wen *et al.* introduced PermOF, an accurate authorization mechanism that allows for restricted access to OpenFlow-enabled SDN as well as data path to OpenFlow-based applications. The proposed solution in [154] establishes controlled access according to a combination of isolation and authorization techniques (e.g., system-based authorizations) to carry out an authorized and controlled access. However, the proposed security mechanism was not experimentally evaluated.

As the implemented applications in SDN controller need to fulfill the global network view to keep up with the network updates and changes, Beckett *et al.* in [225], suggest a mechanism to check, belay and rectify SDN-based applications to keep up (e.g., be alerted) with the network changes. The proposed assertion-based mechanism grants a program bugs detection. It utilizes a probing language to allow network operators to examine and rectify dynamic attributes and characteristics of SDN applications.

Flover [226] is an OpenFlow checking system-based framework to assure the flow policies aggregations do not breach the security policies of a particular network. Flover is designed as an OpenFlow-based application running on the SDN to ensure flow rules inserted by the controller are harmonious with the specified properties. Further, VeriFlow, which is presented in [199], is a dynamic mechanism behaving as an inter-layer between the networking devices and the SDN controller. It examines and verifies the network configuration dynamically at run time. The proposed solution facilitates finding bugs using an incremental data structure-based algorithm with minimal impact on network performance.

Other solutions such as [227] and [229] provide an autonomous testing mechanism to find OpenFlow-based applications bugs. In [229], Handigol *et al.* proposed a platform to rectify network applications and identify the root cause of application bugs. Like [229], OFRewind [230] is another security platform to identify anomalous network behaviors. OFRewind permits logging and replaying particular traffic to identify abnormal traffic and activities in a networking environment.

Moreover, Li *et al.* [168] presented CCD, a covert channel defender solution to thwart rule conflicts generated by

TABLE 7. Implementations and solutions for application layer security enhancement.

Solution	Targeted Threat	Solution Type
Fresco [224]	Threat within and/or from applications	Framework for security applications development
PermOF [154]	Access control	Applications control system
Assertion [225]	Flow rules contradiction	Applications debugging framework
Flover [226]	Security policy violation	Security policy verification applications
OFTesting [227]	Faulty programs	Applications testing framework
Leal et al. [228]	Malicious user rules	Flow-based framework for early detection of malignant user rules
Li et al. [168]	Applications rule conflict	Covert channel defense mechanism for resolving rule conflicts and malicious rules insertion

SDN-enabled applications. Li *et al.* [168] first investigated the rule conflict problem before detecting covert channel attacks caused by policy and rule conflicts in applications. The proposed CCD [168] protects the SDN infrastructure from covert channel attacks by validating and resolving rule conflicts. The proposed CCD traces modification messages and rule injection by SDN applications. CCD also analyzes the correlation pattern within inserted rules using fields of the packet header and solves malicious rules conflict in real-time before they are installed.

Although several solutions have been proposed to assist with designing and implementing security applications in SDN about protecting the SDN control layer from anomalous and threatening applications, there is still a scarce potential effort to improve the SDN applications' security. The remaining significant challenges can be summarized as follows:

- There is no proposed solution to distinguish between the user applications, third-party applications, or even service-based applications.
- accountability mechanisms and access control techniques for nested SDN applications are not proven efficient yet.

Last, to summarize the key points, Table 9 gives a taxonomic overview of security solutions and platforms discussed in this section for each SDN layer, control, data, and application layers.

D. THREAT DETECTION & PREVENTION TECHNIQUES

SDN allows the controller to be in an isolated network segment, and we have the management interface in the management network and the data communicated through the data network. However, even though we have the flexibility of isolating them, we have a high-value target in operating the controller to control the entire network. The attack surface targeting the SDN network could either attack the data through the data plane or the control signals through the control plane. However, the SDN controllers are also exposed to threats and attacks via the data plane. In SDN, it is possible for the attackers to traverse from the data plane to the control plane. For instance, when the OpenFlow protocol is used in SDN controllers to control switches, it facilitates the flow rules communication and data packet processing (e.g., checks incoming packets against all rules in the flow table) and switching. If it does not match any of the rules provided by the controller, it forwards a copy of that packet to the SDN controller and requests its assistance. In such scenarios, if there are some vulnerabilities in the processing logic of the SDN controller, then we cannot reverse from the data plane to the control plane. Table 8 summarizes the list of threats in SDN application and their detection and prevention techniques.

For mitigating the threats in SDN architectures, the authors in [231] developed an algorithm to thwart the DoS and malware attacks, and it was tested and evaluated on multiple adversarial settings. In [39], the authors surveyed the SDN-based intrusion detection and prevention mechanisms from the context of security in IoT services, which was incorporated based on the manufacturer usage description. Han *et al.* [40] summarized the possible security threats and possible solutions for mitigating them in SDN controllers, primarily focusing on the threat impacts on the SDN control layer. In one of the recent works by Yurekten *et al.* [41], the authors presented SDN-based cyber defense types, techniques, strategies, and their deployment for assessing the impact of common attacks.

A fog-assisted SDN-driven intrusion detection and prevention system was developed by the authors in [232] for anomaly detection in IoT networks. Further, the solution granted mitigation of the identified threats with minimum computation resources compared to traditional approaches. Malik *et al.* [233] used a hybrid deep learning approach for the implementation of timely and efficient detection of multi-vector attacks and threats in SDN. Here, the evaluation was performed to detect accuracy and analyze the speed as well as efficiency of using deep neural networks in SDN. Ahmad *et al.* [234] summarized the impact and benefits of ML techniques for securing the SDN from DoS and DDoS attacks, which addresses the vulnerabilities in fingerprinting security.

Node replication attacks or clone attacks are more prominent in wireless sensor networks (WSNs), where the attackers make multiple nodes with the same identity and deploy them at various places across the network. In [235], the authors proposed a hybrid clone node detection mechanism enabled through SDN to protect the WSN. This helps thwart attacks by removing the clones occupying the network, thereby improving the performance through better detection mechanisms. Bhayo *et al.* [236] proposed an SDN-based time-efficient and secure IoT framework for addressing the DDoS attacks, which was carried out by analyzing various classes

of parameters observed from the large volume of traffic in the SDN communication. Further, it was also reported to provide higher detection accuracy with lower false positive rates. Similarly, in [237], the DDoS attacks in IoT networks are addressed using the SDN-based honeypots. It was implemented using the moving target defense architecture, where the SDN-based honeypots mimic the IoT devices and safeguard the original IoT devices from malware and attackers.

TILAK [238] was introduced as a token-based prevention technique for the detection of possible threats in SDN. Here, the flooding, poisoning, and replay attacks of the layer discovery protocol were comprehensively analyzed. Further, this approach also confirms mitigating the threats that possess lower resource penalties. The authors in [239] deployed a machine learning (ML)-enabled threat detection mechanism for securing SDN-based networks, which was also experimented with mitigating multiple attacks under various scenarios, thereby providing secure data sharing in the network. A hybrid deep neural network architecture was developed [240] for SDN orchestration targeted toward combating the impact of cyber threats on the Internet of Medical Things (IoMT). Here, the testing efficiency and detection accuracy are assessed and observed to provide efficient and timely detection and mitigation of malware in the network.

E. SECURITY AUTOMATION AND REACTION IN VIRTUALIZED NETWORKS

In complex and large-scale networks, manual security operations may significantly delay or even hinder the detection and mitigation of increasing security attacks [248]. Therefore, as SDN's primary benefit is to facilitate the management and operation of networks with reduced human intervention, security automation in these networks using SDN technology has become intrinsic [249].

Security automation in SDN-enabled environments can be categorized into several complexity levels. The automation level can be measured based upon key qualitative properties; self-optimization, self-healing, self-configuration, and self-adaptation. Further, the complexity automation level can be measured based on two parameters such as, implementation requirements and the amount of storage and processing resources [250].

Automated security and policy tools have been remarkably studied and explored to enhance network-based security services and minimize human interventions and errors. Moreover, the agility needed for security automation tools can be facilitated by integrating SDN & NFV capabilities and transitioning network security management from the hardware to the software level. Notable works such as [251] present an SDN framework to optimally and autonomously allocate and configure security features in virtualized networks. The proposed framework provides a formal verification of the security functions and direct integration into cloud orchestrators.

The authors in [252] developed a zero-touch framework for security automation and orchestration through SDN & NFV. The proposed framework addresses security functions and policy configuration in SDN-enabled UAV systems. It optimizes security orchestration based on a broad range of contextual factors associated with software and hardware conditions.

SDN and NFV technologies introduce modern security enablers to IoT and industrial networks, providing them with higher flexibility and scalability degrees needed to cope with the ever-increasing security automation and configuration requirements [253]. Many studies in the literature considered integrating or leveraging SDN infrastructure to enhance security in industrial and IoT networks [254]. However, only a few deploy SDN and NFV technologies to improve or automate security configurations in these networks. Notably, the authors in [255] specifically enhanced the IoT honeynets using SDN and NFV technologies by optimizing the security automation process. The proposed solution adopts a security policy-based mechanism to enforce honeynets orchestration and facilitate IoT network management. The authors in [256] propose an SDN architecture to enhance security automation in smart grids. The architecture comprises three layers, risk assessment, threat detection, and self-healing to dynamically evaluate the threat level, detect and correlate threat events, and thwart the potential threats.

Although security and policy automation tools for virtualized networks are feasible and help thwart cybersecurity threats, there are limited set of automatic security orchestration tools providing a direct integration into SDN-supported cloud orchestrators with minimal human intervention. Moreover, comprehensive automation architectures fulfilling detection and mitigation of multi-vector attacks, mutual trustworthiness of entities in proportionally unknown topologies, access control and identity management are still missing in virtualized SDN systems [257].

V. LEARNED LESSONS AND OPEN CHALLENGES

In this survey, we have covered the recent contributions of SDN research made toward secure communication systems. While securing the SDN has excellent potential in establishing a robust communication infrastructure, it is not feasible without its inherent limitations that need to be addressed. In this section, we summarize and reiterate the insight we have explored among various areas of SDN that impact the security of SDN communication systems, as discussed in this survey. The reviewed research contributions utilize the inherent virtues of the infrastructure and its defense mechanisms in these areas of securing SDN communication and specific mechanisms discussed in previous sections.

A. THE SCALE OF SDN COMMUNICATION SYSTEMS

A clear lesson learned from the recent works is that security threats for SDN systems are in almost all infrastructural layers. Moreover, based on the scale of the network, there is a proportionate increase in the probability of the attacks. Risk assessment of safety data link and SDN network communication requires adaptive quantitative schemes and frameworks. Further, fault-tolerant algorithms ensure the reliability

TABLE 8. SDN threat detection and prevention techniques.

References	Threat Detection Application	Prevention technique
Yurekten et al. [41]	Common attacks	SDN-based cyber defense
Shafi et al. [232]	Anomaly detection in IoT networks	Fog-assisted SDN driven intrusion detection
Malik et al. [233]	Multi-vector attacks and threats	Hybrid deep learning approach
Ahmad et al. [234]	DoS and DDoS attacks	ML techniques in fingerprinting security
Devi et al. [235]	WSN attacks	Hhybrid clone node detection
Bhayo et al. [236]	DDoS attacks	Time-efficient secure IoT framework
Luo et al. [237]	DDoS attacks in IoT networks	SDN-based honeypots
Nehra et al. [238]	Flooding, poisoning and replay attacks	Token-based
Sebbar et al. [239]	Mitigate multiple attacks	ML-enabled prevention
Liaqat et al. [240]	IoMT	Mitigation of malwares

TABLE 9. Taxonomy of security solutions and platforms for control, data, and application layers in SDN.

Security Platform	Application	Targeted Layer
FRESCO [224]	Anomaly detection and mitigation	Control
PermOF [154]	Authorization control of applications	Control, data
Assertion [225]	Flow rules inspection applications debugging	Application, data
VeriFlow [199]	Flow rules checking and debugging	Data
Flover [226]	Flow rules and policy checking and OpenFlow applications debugging	Control, data, application
OFTesting [227]	Verification and debugging of SDN applications	Application
SE-Floodlight [171]	Applications authorization and security audit	Control, Data
DDoSDetection [178], [198], [241]	DDos detection through SOM	Control, data
HyperFlow [48]	Availability in SDN controller	Control
Min-Cut Placement [222]	Reliability in SDN controller	Control, data
Kempf et al. [204]	Monitoring of infrastructure layer connectivity	Data
FortNOX [209]	Applications permission and flow rules checking	Control, data
FlowChecker [167]	Inspection of malicious configurations	Control, data
DISCO [173], [174]	Availability and traffic monitoring in SDN	Control
DCP [201]	Availability and scalability in SDN	Control
Ctrl-Placement [175]	Availability and scalability in SDN	Control
Ctrl-Reliability [176], [177]	SDN availability	Control
POCO [202]	Failure tolerability and SDN resilience enforcement	Control
Resonance [210]	Policy enforcement and access control	Data
CPRecovery [211]	DDoS and switch connectivity threats	Control, data
Varga et al. [242]	FPGA-based DDoS prevention	Control, data
SDN-Guard [164]	Framework for SDN rootkits detection	Control
SecControl [243]	Network protection framework combining various existing security tools	Control, data, application
Park et al. [244]	ML-based IDPS for botnet attack	Control, data
JESS [245]	Joint entropy-based IDPS for DDoS attack	Control, data
RE-CHECKER [165]	RESTful services and policies verification	Control, data, application
SECOD [207] [208]	IDPS for DoS	Control, data
TENNISON [246]	Multi-level distributed monitoring and remediation framework	Control, data, application
FloodShield [130]	IDPS for data-control saturation	Control, data
CCD [168]	Covert channel defender	Control, application
Ujcich and Sanders [247]	Data protection intent	Control, data, application

of SDN data by assessing the safety of data links and network risks. Also, in large-scale SDN communication systems, the performance analysis for multi-priority data flow scheduling is crucial for guaranteed risk assessment of safety-critical data communication in the digital safety feature control systems. Moreover, intelligent traffic management and load balancing can be achieved by predicting the packet flow in the network, helping in reducing the congestion in the SDN communication scenario.

B. INTELLIGENT AND SECURED SDN CONTROLLERS

The centralized means of SDN control facilitates network programmability for adaptive and automated control. However, intelligence in forwarding and processing incoming packets at a single point could be addressed through various deployment models that ensure secure handling and processing of data. The secured SDN controllers can enable control through physically centralized, distributed, or hybrid architectures. Moreover, various performance parameters and security measures must be balanced without compromising consistency, scalability, and reliability. The intelligent SDN controllers should provide a trustworthy forwarding of data in the communication system and adequately address all the performance parameters with the secure communication framework. Further, networking, operating, and accessing the resources in the cloud environment through the secure SDN services need additional intelligence at the controller with the compromise on the performance parameters.

C. RESEARCH AND PRACTICAL IMPLICATIONS

1) NETWORK POLICIES

The worth of SDN lies in its capability to guarantee coherent policy enforcement and better scalability due to its centralized management and network programmability [31]. The future generation of security solutions will benefit from the richness of network usage information available in SDN to enhance security policies enforcement, network anomaly revelation, and attenuation.

2) CONTROLLER COMPROMISE

In [29], a survey was conducted on how to improve network security using the characteristics of the SDN architecture, and what vulnerabilities and security issues the SDN architecture may introduce. However, the existing knowledge on SDN attacks is quietly limited. The current systems and solutions imposed on carrying out SDN functions may be prone to malignant security threats. Adversaries will inevitably exploit these SDN systems if a successful network compromise is feasible through exploiting SDN infrastructure vulnerabilities. A susceptible SDN environment could therefore undermine the security and availability of the entire networking system [26].

3) THREAT ANTICIPATION

To secure SDN environments, all of the potential security threats need to be anticipated before adversaries exploit their vulnerabilities [198], [258]. Moreover, an efficient threat mitigation model has to consider looking at SDN from the attacker's perspective to highlight potential threats/anomalies on SDN at the architectural level, regardless of whether these threats can be successfully carried out.

4) INTRUSION DETECTION AND PREVENTION

In an SDN infrastructure, the mitigation scheme aims to resist a certain attack(s) on networks attached to the Internet (i.e., interconnecting) network by securing the target and relay networks. This could be achieved by passing network traffic to the attacked network through high capacity networks with traffic scrubbing filters (e.g., DDoS mitigation demands an appropriate identification of incoming traffic to distinguish human-like bots from human traffic and hijacked web browsers) [209]. Based on the threat model, risk speculation can be further elaborated to define what threats are likely to pose realistic dangers to a specific SDN deployment, thus, designing a more feasible mitigation scheme. For each attack to be mitigated, security requirements should be identified, outlining a group of goals to attain the desired SDN security. Hence an optimal way of threat mitigation using the benefits of SDN infrastructure needs to be elaborated.

5) LACK OF TLS ADOPTION

TLS and Datagram Transport Layer Security (DTLS) are quietly complex to implement and configure in SDN-enabled devices from the technical perspective [138] (e.g., requires the creation of global certificates including SDN and OVS devices certificates, generation of appropriate keys and certificates for all networking devices in the SDN environment, etc.). Hence, networking operators and vendors have ignored TLS deployment in OpenFlow switching devices and made it optional. However, even with the adoption of TLS specifications in an SDN-enabled environment, the OpenFlow switches are still vulnerable to TCP level threats due to the lack of TCP level protection in TLS implementations. In [138] and [259], Benton *et al.* and Dierks highlight TLS as well as DTLS specifications for OpenFlow protocol deployment. Nevertheless, the recent OpenFlow versions (i.e., version 1.3 and above) do not enforce TLS adoption and render it optional.

6) CONTROL PLANE PERSPECTIVE

As the SDN controller provides logically centralized-based control decisions through its control plane, this renders the entire SDN infrastructure vulnerable and hence targeted when proceeding with anomalous networking actions and activities.

7) DATA PLANE PERSPECTIVE

In SDN technology, if OpenFlow switching devices do not extradite any instructions about flow forwarding rules from the SDN controller, this implies that the SDN control layer is disconnected or there is a control layer failure at some point. Therefore, this scenario renders the link off-line and allows for its exploitation by an adversary that could insert new flow rules to serve their goals. The separation of the management layer from the infrastructure layer can present new threats such as teleportation, where an adversary attempts to send information via the control layer bypassing pre-defined network functions at the data layer level [163]. Thus, the security of the control layer has an explicit effect on the data layer [160]. This implies that if an SDN controller is compromised, all data layer entities will be compromised.

8) APPLICATION PLANE PERSPECTIVE

The application plane in SDN is decomposed of implemented applications, which interactively program the network with the control plane of SDN. These various applications could be autonomous and possessed by multiple network users.

As SDN fulfills the networking foundation specifications, it guarantees to manage the entire networking environment by software through logically centralized control mechanisms [153]. However, SDN architecture does not provide mechanisms and specifications to expedite the use of open APIs for applications to manage, control, and monitor network services and features via the SDN control layer [155]. Hence, this limitation renders the SDN infrastructure and networking devices and resources vulnerable to attacks by malicious applications. These security challenges are evolving while the OpenFlow protocol does not offer any compulsory security mechanisms to enforce the access control and authentication for implemented SDN-enabled applications.

D. FUTURE RESEARCH TRENDS

The SDN controller allows developers and network administrators to implement advanced and efficient networking architectures, models, and operational network applications. This flexibility eventually carries out creative inventions as well as presents security threats, and challenges in the networking industry and research [260]. This subsection discusses open research problems and future research opportunities for secure integration of SDN architecture in smart city communication systems. The key research directions are summarized and discussed in detail as follows.

- Examination of the controller software implementations prior to integration into smart city communication systems to identify possible exposures to common pitfalls and design vulnerabilities.
- Exploration of policy integration complexity and policy collision in the distributed SDN control plane.
- Enforcement of authorization and access control of SDN-enabled applications according to the distinguished operations' demands while preserving the networking overhead constraints.
- Enhancement of scalability to prevent adversaries from elaborating on attacks based on the immersing controller-to-data channel communication.
- Addressing the cascading deficiency caused by multi-SDN controllers deployment.
- Enhancement in SDN-enabled business intelligent decision making and imparting robust security shield through intent-based networking (IBN) and blockchain deployment.

1) IMPLEMENTATION

A broad range of OpenFlow implementations have been examined to investigate their exposure to common sets of pitfalls and design weaknesses (e.g., [261]), which allow for an intensive amount of security threats. Thus, the SDN-enabled independent applications may utilize the functionalities of various SDN elements at a time, introducing severe security vulnerabilities. Besides, when an SDN application (i.e., whether a user or administrator-based) is implemented in the control plane in a detached networking environment, the SDN can be prone to further security challenges, such as policy integration complexity and policies collision.

The vast majority of networking operations are perceived to be installed as networking-enabled applications in software within the SDN control plane (i.e., control-application layer level). While particular implementations in SDN software may require network statistics about load-balancing, other applications could require flow samples, etc. Thus, each particular type of SDN application must have a reasonable and safe authorization and access control according to its specific operation demands in order to maintain a determined jurisdiction and utilize a reliable traffic route while preserving the networking overhead constraints.

Furthermore, a categorization of applications impacting the SDN resiliency is needed based upon specific criteria; packaged services of network, services for the network system, and critical networking applications. Authorization and access control mechanisms in this context should not be unified for all SDN applications; otherwise, the control layer may experience a bottleneck as a result of the large quantity of arriving requests to gain entry to networking elements and resources.

2) SCALABILITY

Scalability is another considerable challenge in the centralized SDN controller, where the quantity of control flows augments as the topology scale increases. As a result, the response time of the flow rules setup significantly increases. Furthermore, the scarcity of SDN scalability can allow adversaries to establish attacks based on the immersing controller-to-switch communication to saturate the SDN control layer, thus compromising the exhausted switching devices. Although several studies proposed the employment of multiple SDNs to resolve the availability challenges in SDN, such a mechanism typically leads to a cascading deficiency. Therefore, the corresponding scalability to SDN resiliency must be taken into consideration in order to grant reliable SDN availability.

3) CONTROLLER AND SWITCH RESPONSE TIME

The SDN controller offers control and application layer-based services for a broad range of traffic forwarding entities. Such an SDN capability can lead to a controller-to-switch and switch-to-switch latency increase when reciprocating the network state and resources inquiries, introducing new vulnerabilities related to SDN availability. It is also feasible that the larger the number of connected switching devices becomes, the higher the controller response time of installing traffic rules is since more incoming traffic requires more setup demands from the controller. Hence, a smart trade-off among the infrastructure and control layers is recommended as an eventual solution to optimize the switch reliance on the SDN and improve scalability and delay through internal decision-making abilities.

4) POLICIES DEPLOYMENT

The security of a network environment is a key structural component of network management, and resilient policy adoptions demand a comprehensive analysis of policies' configurations in order to avert policy conflicts, minimize the risks of security vulnerabilities, and maintain the network flows alive when a security breach occurs. Like in traditional networks, flow characteristics, features, and statistics are deployed to capture flooding threats. Although several studies (e.g., [178], [198], [262]) addressed the control-to-data layer saturation attacks in reactive controllers via lightweight implementations for independent detection and mitigation mechanisms. However, SDN's holistic and centralized networking view and its infrastructure layer's flexible

programmability are likely to allow for interdependent and mutual policies deployment. Thus, it is recommended to design interdependent policies for security and flow forwarding that guarantee a secure forwarding of data flows.

5) AUTOMATED RECOVERY

SDN further allows for introducing languages and controllers that have the ability to react under the network state alterations dynamically. SDN controllers provide a framework for efficient automation and monitoring of the network state, rendering the design of new tasks in automation-based applications simple. Consequently, the communication and SDN operations costs must be minimized through dedicated automation mechanisms. Such mechanisms can be designed and developed based on platforms dedicated to automated policies and autonomous control implementations. However, no practical mechanism for policy automation has been tested in SDN yet.

The logical centralization of the SDN brings in more charges for network operators as the paucity of the operator's awareness, and familiarity could render the networking environment prone to bottleneck threats. Thus, autonomous recovery applications and automated, flexible, and advanced security mechanisms are recommended to be placed on top of the SDN controller so that operators only need to provide minimal involvement to secure the communication system.

6) FLOW TRACKING

Future network security defense mechanisms are recommended to extend current tracking techniques of data flows on each host for specific network-level defenses. As such, the SDN paradigm may be leveraged with data flow tracking techniques to capture a broad range of cyber threats utilized by knowledgeable attackers targeting both network and application layer protocols. Current network traffic monitoring and attack detection practices still mainly rely on simple statistics, suffering from poor performance and low accuracy. Future developments can employ more advanced ML techniques for analyzing large-scale testbed traffic data as it is currently challenging to scale to the volume, velocity, and complexity of the traffic data from large-scale networks.

7) BUSINESS INTELLIGENCE

To meet the business demands enabled through SDN technology and provide effective network management, the network can be made more intelligent through IBN. This framework allows us to capture the business intent, check the integrity, and translate it to policies. Further, continuous network alignment will be guaranteed with assurance on continuous verification, insights, and corrective actions through AI-enabled service assurance capabilities in the network. The primary role of SDN in the IBN is to orchestrate the policies and automate secured system configuration targeted for QoE-driven business models [263].

8) RESILIENCY

SDN-based communication systems are vulnerable to different network resiliency incidents, including energy outages, connectivity disruption, and controller crashes. Such incidents are more likely to occur when the network is under specialized attack scenarios. The vast majority of existing research studies in SDN security focus on predicting, preventing, detecting, and mitigating attacks, but limited number studies address network state recovery after successful attacks. Complete security solutions can be developed based upon a combination of proactive and active approaches to simultaneously act as a front-line shield against security attacks and enforce the network state recovery once the SDN infrastructure is successfully compromised.

9) BLOCKCHAIN INTEGRATION

Lastly, a further future direction is the integration of blockchain technology and SDN into smart city applications. An integration example can be blockchain as-a-Service [264]. In this direction, a permissioned blockchain can be deployed to prevent malware injection against not only the SDN planes, but also the intermediate communication paths. Further, efficient authentication methods can be incorporated through distributed trust services, ensuring lower energy consumption with lower delay and improved throughput in the SDN routing decisions.

VI. CONCLUSION

SDN architecture provides a flexible and agile experience for the end-users in a wide range of applications. The SDN market has evolved in response to the demands from large data centers toward the aggregation of multiple types of network connections. SDN has further enabled solutions for high-demand resources, managing unpredictable data traffic patterns, and rapid network reconfiguration. It is used to virtualize the network by separating the control plane from the data plane, where data flows are separated from control flows. However, the rapid increase in the number of smart devices connected to the network has increased the data traffic in the network and raised security issues in SDN-enabled communication systems. In this paper, we conducted a comprehensive survey on the core functionality of SDN from the perspective of secure communication at different scales. A specific focus is given to address the challenges of securing SDN infrastructure. We further categorized the appropriate solutions for specific threats at each layer of the SDN communication framework. Lastly, security implications and future open research challenges are presented to help the community gain further insights into the domain of SDN security.

ACKNOWLEDGMENT

The views and conclusion contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of NSF.

REFERENCES

[1] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, *Software-Defined Networking (SDN): Layers and Architecture Terminology*, document RFC 7426, 2015.

[2] ETSI. *Network Functions Virtualisation (NFV)*. Accessed: Feb. 2022. [Online]. Available: https://www.etsi.org/technologies/nfv

[3] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2019.

[4] A. Abdou, P. C. van Oorschot, and T. Wan, "A framework and comparative analysis of control plane security of SDN and conventional networks," 2017, *arXiv:1703.06992*.

[5] M. Rahouti, K. Xiong, Y. Xin, and N. Ghani, "QoSP: A priority-based queueing mechanism in software-defined networking environments," in *Proc. IEEE Int. Perform., Comput., Commun. Conf. (IPCCC)*, Oct. 2021, pp. 1–7.

[6] M. Rahouti, K. Xiong, Y. Xin, and N. Ghani, "A priority-based queueing mechanism in software-defined networking environments," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2021, pp. 1–2.

[7] K. Sood, K. K. Karmakar, S. Yu, V. Varadharajan, S. R. Pokhrel, and Y. Xiang, "Alleviating heterogeneity in SDN-IoT networks to maintain QoS and enhance security," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5964–5975, Jul. 2020.

[8] X. Yang, D. Wang, W. Tang, W. Feng, and C. Zhu, "IPsec cryptographic algorithm invocation considering performance and security for SDN southbound interface communication," *IEEE Access*, vol. 8, pp. 181782–181795, 2020.

[9] A. Sallam, A. Refaey, and A. Shami, "On the security of SDN: A completed secure and scalable framework using the software-defined perimeter," *IEEE Access*, vol. 7, pp. 146577–146587, 2019.

[10] A. Derhab, M. Guerroumi, A. Gumaei, L. Maglaras, M. A. Ferrag, M. Mukherjee, and F. A. Khan, "Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security," *Sensors*, vol. 19, no. 14, p. 3119, 2019.

[11] J. H. Cox, R. Clark, and H. Owen, "Leveraging SDN and WebRTC for rogue access point security," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 756–770, Sep. 2017.

[12] M. Tiloca, A. Stagkopoulou, and G. Dini, "Performance and security evaluation of SDN networks in OMNeT++/INET," 2016, *arXiv:1609.04554*.

[13] S. N. Matheu, A. R. Enciso, A. M. Zarca, D. Garcia-Carrillo, J. L. Hernández-Ramos, J. B. Bernabe, and A. F. Skarmeta, "Security architecture for defining and enforcing security profiles in DLT/SDN-based IoT systems," *Sensors*, vol. 20, no. 7, p. 1882, Mar. 2020.

[14] A. Abdou, P. C. van Oorschot, and T. Wan, "Comparative analysis of control plane security of SDN and conventional networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3542–3559, 4th Quart., 2018.

[15] H. Zhong, J. Sheng, Y. Xu, and J. Cui, "SCPLBS: A smart cooperative platform for load balancing and security on SDN distributed controllers," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 440–451, Mar. 2019.

[16] N. Dayal, P. Maity, S. Srivastava, and R. Khondoker, "Research trends in security and DDoS in SDN," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6386–6411, Dec. 2016.

[17] R. Sahay, W. Meng, D. A. S. Estay, C. D. Jensen, and M. B. Barfod, "CyberShip-IoT: A dynamic and adaptive SDN-based security policy enforcement framework for ships," *Future Gener. Comput. Syst.*, vol. 100, pp. 736–750, Nov. 2019.

[18] R. Sultana, J. Grover, and M. Tripathi, "Security of SDN-based vehicular ad hoc networks: State-of-the-art and challenges," *Veh. Commun.*, vol. 27, Jan. 2021, Art. no. 100284.

[19] M. Arif, G. Wang, O. Geman, V. E. Balas, P. Tao, A. Brezulianu, and J. Chen, "SDN-based VANETs, security attacks, applications, and challenges," *Appl. Sci.*, vol. 10, no. 9, p. 3217, May 2020.

[20] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A comprehensive survey: Benefits, services, recent works, challenges, security, and use cases for SDN-VANET," *IEEE Access*, vol. 8, pp. 91028–91047, 2020.

[21] H. Shafiq, R. A. Rehman, and B.-S. Kim, "Services and security threats in SDN based VANETs: A survey," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–14, Apr. 2018.

[22] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *Proc. IEEE SDN Future Netw. Services (SDNFNS)*, Nov. 2013, pp. 1–7.

[23] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 4th Quart., 2014.

[24] G. Garg and R. Garg, "Review on architecture and security issues in SDN," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 6519–6524, 2014.

[25] H. Farhady, L. HyunYong, and N. Akihiro, "Software-defined networking: A survey," *Comput. Netw.*, vol. 81, pp. 79–95, Apr. 2015.

[26] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, 4th Quart., 2015.

[27] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[28] S. Bian, P. Zhang, and Z. Yan, "A survey on software-defined networking security," in *Proc. 9th EAI Int. Conf. Mobile Multimedia Commun.*, 2016, pp. 190–198.

[29] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[30] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," *J. Netw. Comput. Appl.*, vol. 67, pp. 1–25, May 2016.

[31] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, and H. L. Owen, "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25487–25526, 2017.

[32] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti, "A survey on the security of stateful SDN data planes," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1701–1725, 3rd Quart., 2017.

[33] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2456–2501, 4th Quart., 2017.

[34] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, Jan. 2019.

[35] K. Nisar, I. Welch, R. Hassan, A. H. Sodhro, and S. Pirbhulal, "A survey on the architecture, application, and security of software defined networking: Challenges and open issues," *Internet Things*, vol. 12, Dec. 2020, Art. no. 100289.

[36] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, "Software-defined network (SDN) data plane security: Issues, solutions, and future directions," in *Handbook of Computer Networks and Cyber Security*. Cham, Switzerland: Springer, 2020, pp. 341–387.

[37] A. Hussein, L. Chadad, N. Adalian, A. Chehab, I. H. Elhajj, and A. Kayssi, "Software-defined networking (SDN): The security review," *J. Cyber Secur. Technol.*, vol. 4, no. 1, pp. 1–66, 2020.

[38] J. C. C. Chica, J. C. Imbachi, and J. F. B. Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, Jun. 2020, Art. no. 102595.

[39] N. Mazhar, R. Salleh, M. Asif, and M. Zeeshan, "SDN based intrusion detection and prevention systems using manufacturer usage description: A survey," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 717–737, 2020.

[40] T. Han, S. R. U. Jan, Z. Tan, M. Usman, M. A. Jan, R. Khan, and Y. Xu, "A comprehensive survey of security threats and their mitigation techniques for next-generation SDN controllers," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 16, p. e5300, 2020.

[41] O. Yurekten and M. Demirci, "SDN-based cyber defense: A survey," *Future Gener. Comput. Syst.*, vol. 115, pp. 126–149, Feb. 2021.

[42] *Floodlight Controller*. Accessed: Feb. 2022. [Online]. Available: http://www.projectfloodlight.org/floodlight/

[43] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd ACM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2014, pp. 1–6.

[44] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.

[45] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.

[46] *Ryu: SDN Framework*. Accessed: Feb. 2022. [Online]. Available: https://osrg.github.io/ryu/

[47] *OpenFlow Controller: SNAC (Simple Network Access Control)*, Stanford Univ. Big Switch Networks.
[48] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, 2010, p. 3.
[49] *OpenMUL Controller*. Accessed: Feb. 2022. [Online]. Available: http://www.openmul.org/
[50] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 19–24.
[51] *Opencontrail*. Accessed: Feb. 2022. [Online]. Available: http://www.opencontrail.org
[52] *Trema: A Full-Stack OpenFlow Framework in Ruby and C*. Accessed: Feb. 2022. [Online]. Available: https://trema.github.io/trema/
[53] D. Erickson, "The beacon OpenFlow controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 13–18.
[54] S. Kaur, J. Singh, and N. S. Ghumman, "Network programmability using pox controller," in *Proc. IEEE Int. Conf. Commun., Comput. Syst. (ICCCS)*, vol. 138, Aug. 2014, p. 70.
[55] J. H. Cox, S. Donovan, R. J. Clarky, and H. L. Owen, "Ryuretic: A modular framework for Ryu," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 1065–1070.
[56] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Oct. 2010.
[57] J. Reich, C. Monsanto, N. Foster, J. Rexford, and D. Walker, "Modular SDN programming with pyretic," USENIX, Berkeley, CA, USA, Tech. Rep., 2013.
[58] C. Price, S. Rivera, A. Peled, M. Wolpin, F. Brockners, P. Chinnakannan, A. Sardella, P. Hou, M. Young, P. Mehta, T. Nguyenphu, and D. Neary, "OPNFV: An open platform to accelerate NFV," A Linux Found. Collaborative Project, White Paper, 2012. [Online]. Available: https://networkbuilders.intel.com/docs/OPNFV_WhitePaper_Final.pdf
[59] M. Rahouti, K. Xiong, and Y. Xin, "Secure software-defined networking communication systems for smart cities: Current status, challenges, and trends," *IEEE Access*, vol. 9, pp. 12083–12113, 2021.
[60] M. Rahouti, K. Xiong, T. Chin, P. Hu, and D. De Oliveira, "A preemption-based timely software defined networking framework for emergency response traffic delivery," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun., IEEE 17th Int. Conf. Smart City, IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 452–459.
[61] M. Rahouti, K. Xiong, N. Ghani, and F. Shaikh, "SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks," *IET Netw.*, vol. 10, no. 2, pp. 76–87, Mar. 2021.
[62] M. Rahouti, K. Xiong, Y. Xin, and N. Ghani, "LatencyS-masher: A software-defined networking-based framework for end-to-end latency optimization," in *Proc. IEEE 44th Conf. Local Comput. Netw. (LCN)*, Oct. 2019, pp. 202–209.
[63] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable OpenFlow-SDN flow control: A survey," *IEEE Access*, vol. 7, pp. 107346–107379, 2019.
[64] S. J. Vaughan-Nichols, "OpenFlow: The next generation of the network?" *Computer*, vol. 44, no. 8, pp. 13–15, 2011.
[65] A. Azzouni, N. T. M. Trang, R. Boutaba, and G. Pujolle, "Limitations of OpenFlow topology discovery protocol," in *Proc. 16th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2017, pp. 1–3.
[66] P. L. Ventre, M. M. Tajiki, S. Salsano, and C. Filsfils, "SDN architecture and southbound APIs for IPv6 segment routing enabled wide area networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1378–1392, Oct. 2018.
[67] S. Merle, J. P. Fernandez-Palacios, O. G. D. Dios, L. Gifre, R. Vilalta, and P. Stritzinger, "Scalable and resilient network traffic engineering using erlang-based path computation element," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2021, pp. 100–101.
[68] A. S. Monge and K. G. Szarkowicz, *MPLS in the SDN Era: Interoperable Scenarios to Make Networks Scale to New Services*. Sebastopol, CA, USA: O'Reilly Media, 2015.
[69] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Abstraction and control of multi-domain disaggregated optical networks with OpenROADM device models," *J. Lightw. Technol.*, vol. 38, no. 9, pp. 2606–2615, May 1, 2020.
[70] G. G. Patrushev and V. G. Drozdova, "Routing efficiency evaluation with SDN solutions integration in the data network," in *Proc. 18th Int. Conf. Young Spec. Micro/Nanotechnol. Electron Devices (EDM)*, 2017, pp. 190–194.
[71] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.
[72] B. Niu, J. Kong, S. Tang, Y. Li, and Z. Zhu, "Visualize your IP-over-optical network in realtime: A P4-based flexible multilayer in-band network telemetry (ML-INT) system," *IEEE Access*, vol. 7, pp. 82413–82423, 2019.
[73] R. Farahani, "CDN and SDN support and player interaction for HTTP adaptive video streaming," in *Proc. 12th ACM Multimedia Syst. Conf.*, Jun. 2021, pp. 398–402.
[74] 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, p. e3622, Mar. 2022.
[75] T. Hu, Z. Zhang, P. Yi, D. Liang, Z. Li, Q. Ren, Y. Hu, and J. Lan, "SEAPP: A secure application management framework based on REST API access control in SDN-enabled cloud environment," *J. Parallel Distrib. Comput.*, vol. 147, pp. 108–123, Jan. 2021.
[76] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, 2020.
[77] Y. Gao, Y. Chen, X. Hu, H. Lin, Y. Liu, and L. Nie, "Blockchain based IIoT data sharing framework for SDN-enabled pervasive edge computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5041–5049, Jul. 2021.
[78] J. Li, J. Cai, F. Khan, A. U. Rehman, V. Balasubramaniam, J. Sun, and P. Venu, "A secured framework for SDN-based edge computing in IoT-enabled healthcare system," *IEEE Access*, vol. 8, pp. 135479–135490, 2020.
[79] C. Li, L. Zhu, W. Li, and Y. Luo, "Joint edge caching and dynamic service migration in SDN based mobile edge computing," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102966.
[80] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107739.
[81] G. N. Kumar, K. Katsalis, P. Papadimitriou, P. Pop, and G. Carle, "Failure handling for time-sensitive networks using SDN and source routing," in *Proc. IEEE 7th Int. Conf. Netw. Softwarization (NetSoft)*, Jun. 2021, pp. 226–234.
[82] W. Kong, M. Nabi, and K. Goossens, "Run-time recovery and failure analysis of time-triggered traffic in time sensitive networks," *IEEE Access*, vol. 9, pp. 91710–91722, 2021.
[83] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3130–3142, Mar. 2021.
[84] N. S. Bülbül, D. Ergenç, and M. Fischer, "SDN-based self-configuration for time-sensitive IoT networks," 2021, *arXiv:2103.01282*.
[85] P. Segeč, M. Moravčik, J. Uratmová, J. Papán, and O. Yeremenko, "SD-WAN–architecture, functions and benefits," in *Proc. 18th Int. Conf. Emerg. eLearning Technol. Appl. (ICETA)*, Nov. 2020, pp. 593–599.
[86] K. Alwasel, D. N. Jha, E. Hernandez, D. Puthal, M. Barika, B. Varghese, S. K. Garg, P. James, A. Zomaya, G. Morgan, and R. Ranjan, "IoTSim-SDWAN: A simulation framework for interconnecting distributed data-centers over software-defined wide area network (SD-WAN)," *J. Parallel Distrib. Comput.*, vol. 143, pp. 17–35, Sep. 2020.
[87] Z. Duliński, R. Stankiewicz, G. Rzym, and P. Wydrych, "Dynamic traffic management for sd-wan inter-cloud communication," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1335–1351, 2020.
[88] H. F. Khazaal, H. K. Al-Abassi, A. M. Al-Sadi, and A. Al-Sherbaz, "Evaluating healthcare system based sd-wan backbone," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 1, pp. 671–680, 2020.
[89] K. Abbas, M. Afaq, T. A. Khan, A. Rafiq, J. Iqbal, I. U. Islam, and W. Song, "An efficient SDN-based LTE-WiFi spectrum aggregation system for heterogeneous 5G networks," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, p. e3943, Apr. 2022.
[90] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-based secure and trustworthy Internet of Things in SDN-enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656–56666, 2019.

[91] K. K. Karmakar, V. Varadharajan, S. Nepal, and U. Tupakula, "SDN-enabled secure IoT architecture," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6549–6564, Apr. 2021.
[92] A. Al Hayajneh, M. Z. A. Bhuiyan, and I. McAndrew, "Improving Internet of Things (IoT) security with software-defined networking (SDN)," *Computers*, vol. 9, no. 1, p. 8, Feb. 2020.
[93] H. Ni, M. Rahouti, A. Chakrabortty, K. Xiong, and Y. Xin, "A distributed cloud-based wide-area controller with SDN-enabled delay optimization," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2018, pp. 1–5.
[94] U. Ghosh, P. Chatterjee, and S. Shetty, "A security framework for SDN-enabled smart power grids," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jun. 2017, pp. 113–118.
[95] A. H. M. Jakaria, M. A. Rahman, and A. Gokhale, "Resiliency-aware deployment of SDN in smart grid SCADA: A formal synthesis model," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1430–1444, Jun. 2021.
[96] H. Mahmood, D. Mahmood, Q. Shaheen, R. Akhtar, and W. Changda, "S-DPS: An SDN-based DDoS protection system for smart grids," *Secur. Commun. Netw.*, vol. 2021, pp. 1–19, Mar. 2021.
[97] H. Peng, Q. Ye, and X. S. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 156–162, Aug. 2019.
[98] S. Garg, K. Kaur, S. H. Ahmed, A. Bradai, G. Kaddoum, and M. Atiquzzaman, "MobQoS: Mobility-aware and QoS-driven SDN framework for autonomous vehicles," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 12–20, Aug. 2019.
[99] K. Kaur, S. Garg, G. Kaddoum, N. Kumar, and F. Gagnon, "SDN-based internet of autonomous vehicles: An energy-efficient approach for controller placement," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 72–79, Dec. 2019.
[100] E. Lyczkowski, C. Sauer, N. Brödner, W. Kiess, and M. Schmidt, "SDN controlled visible light communication clusters for AGVS," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, 2021, pp. 154–159.
[101] R. Mohammadi, R. Javidan, M. Keshtgari, and N. Rikhtegar, "SMOTE: An intelligent SDN-based multi-objective traffic engineering technique for telesurgery," *IETE J. Res.*, vol. 0, pp. 1–11, Mar. 2021.
[102] W. Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, "Mobile edge cloud-based industrial Internet of Things: Improving edge intelligence with hierarchical SDN controllers," *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 36–45, Mar. 2020.
[103] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K.-R. Choo, "An energy-efficient SDN controller architecture for IoT networks with blockchain-based security," *IEEE Trans. Services Comput.*, vol. 13, no. 4, pp. 625–638, Jul. 2020.
[104] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad, and J. Wang, "Blockchain-enabled distributed security framework for next-generation IoT: An edge cloud and software-defined network-integrated approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6143–6149, Jul. 2020.
[105] A. Rahman, M. J. Islam, A. Montieri, M. K. Nasir, M. M. Reza, S. S. Band, A. Pescape, M. Hasan, M. Sookhak, and A. Mosavi, "SmartBlock-SDN: An optimized blockchain-SDN framework for resource management in IoT," *IEEE Access*, vol. 9, pp. 28361–28376, 2021.
[106] P. T. Duy, H. D. Hoang, D. T. T. Hien, N. B. Khanh, and V.-H. Pham, "SDNLog-foren: Ensuring the integrity and tamper resistance of log files for SDN forensics using blockchain," in *Proc. 6th NAFOSTED Conf. Inf. Comput. Sci. (NICS)*, Dec. 2019, pp. 416–421.
[107] A. Guerrero-Pérez, A. Huerta, F. González, and D. López, "Network architecture based on virtualized networks for smart cities," White Papers From Smart Cities Future Kickoff Event, Tech. Rep., 2013.
[108] A. A. Abbasi, A. Abbasi, S. Shamshirband, A. T. Chronopoulos, V. Persico, and A. Pescapè, "Software-defined cloud computing: A systematic review on latest trends and developments," *IEEE Access*, vol. 7, pp. 93294–93314, 2019.
[109] I. Alam, K. Sharif, F. Li, Z. Latif, M. M. Karim, S. Biswas, B. Nour, and Y. Wang, "A survey of network virtualization techniques for Internet of Things using SDN and NFV," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–40, Mar. 2021.
[110] A. Manzalini and A. Stavdas, "The network is the robot," *Commun. Strategies*, vol. 1, no. 96, p. 73, 2014.
[111] Q. Wang, G. Shou, Y. Liu, Y. Hu, Z. Guo, and W. Chang, "Implementation of multipath network virtualization with SDN and NFV," *IEEE Access*, vol. 6, pp. 32460–32470, 2018.
[112] D. Qiang, N. Ansari, and M. Toy, "Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Netw.*, vol. 30, no. 5, pp. 10–16, Sep./Oct. 2016.
[113] Y.-H. Kim, J.-M. Gil, and D. Kim, "A location-aware network virtualization and reconfiguration for 5G core network based on SDN and NFV," *Int. J. Commun. Syst.*, vol. 34, no. 2, p. e4160, 2021.
[114] Z. Cai, A. L. Cox, and T. Ng, "Maestro: A system for scalable OpenFlow control," Tech. Rep., 2010. Accessed: Jan. 2022. [Online]. Available: https://www.cs.rice.edu/ eugeneng/papers/TR10-11.pdf
[115] B. Lee, S. H. Park, J. Shin, and S. Yang, "IRIS: The OpenFlow-based recursive SDN controller," in *Proc. 16th Int. Conf. Adv. Commun. Technol.*, Feb. 2014, pp. 1227–1231.
[116] R. Nagarathna and S. M. Shalinie, "SLAMHHA: A supervised learning approach to mitigate host location hijacking attack on SDN controllers," in *Proc. 4th Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Mar. 2017, pp. 1–7.
[117] S. Khan, A. Gani, A. A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 303–324, 1st Quart., 2017.
[118] A. Dumka, H. L. Mandoria, and A. Sah, "Analysis of issues in SDN security and solutions," in *Innovations in Software-Defined Networking and Network Functions Virtualization*. Hershey, PA, USA: IGI Global, 2018, pp. 217–239.
[119] I. H. Abdulqadder, S. Zhou, D. Zou, I. T. Aziz, and S. M. A. Akber, "Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107364.
[120] Y. Li, Y. Qin, P. Zhang, and A. Herzberg, "SDN-enabled cyber-physical security in networked microgrids," *IEEE Trans. Sustain. Energy*, vol. 10, no. 3, pp. 1613–1622, Jul. 2019.
[121] D. Smyth, S. McSweeney, D. O'Shea, and V. Cionca, "Detecting link fabrication attacks in software-defined networks," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–8.
[122] S. Khan, M. A. Bagiwa, A. W. A. Wahab, A. Gani, and A. Abdelaziz, "Understanding link fabrication attack in software defined network using formal methods," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 555–562.
[123] A. Azzouni, R. Boutaba, N. T. M. Trang, and G. Pujolle, "SOFTDP: Secure and efficient topology discovery protocol for SDN," 2017, *arXiv:1705.04527*.
[124] X. Huang, P. Shi, Y. Liu, and F. Xu, "Towards trusted and efficient SDN topology discovery: A lightweight topology verification scheme," *Comput. Netw.*, vol. 170, Apr. 2020, Art. no. 107119.
[125] R. Skowyra, L. Xu, G. Gu, V. Dedhia, T. Hobson, H. Okhravi, and J. Landry, "Effective topology tampering attacks and defenses in software-defined networks," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2018, pp. 374–385.
[126] T. A. V. Sattolo, S. Macwan, M. J. Vezina, and A. Matrawy, "Classifying poisoning attacks in software defined networking," in *Proc. IEEE Int. Conf. Wireless Space Extreme Environ. (WiSEE)*, Oct. 2019, pp. 59–64.
[127] S. Macwan and C.-H. Lung, "Investigation of moving target defense technique to prevent poisoning attacks in SDN," in *Proc. IEEE World Congr. Services (SERVICES)*, Jul. 2019, pp. 178–183.
[128] D. Smyth, D. O'Shea, V. Cionca, and S. McSweeney, "Attacking distributed software-defined networks by leveraging network state consistency," *Comput. Netw.*, vol. 156, pp. 9–19, Jun. 2019.
[129] J. Hua, Z. Zhou, and S. Zhong, "Flow misleading: Worm-hole attack in software-defined networking via building in-band covert channel," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1029–1043, 2021.
[130] M. Zhang, J. Bi, J. Bai, and G. Li, "FloodShield: Securing the SDN infrastructure against denial-of-service attacks," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 687–698.
[131] E. Marin, N. Bucciol, and M. Conti, "An in-depth look into SDN topology discovery mechanisms: Novel attacks and practical countermeasures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 1101–1114.

[132] B. E. Ujcich, "Securing the software-defined networking control plane by using control and data dependency techniques," Ph.D. dissertation, Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 2020.

[133] A. Pradhan and R. Mathew, "Solutions to vulnerabilities and threats in software defined networking (SDN)," *Proc. Comput. Sci.*, vol. 171, pp. 2581–2589, Jan. 2020.

[134] T. Ubale and A. K. Jain, "Survey on DDoS attack techniques and solutions in software-defined network," in *Handbook of Computer Networks and Cyber Security*. Cham, Switzerland: Springer, 2020, pp. 389–419.

[135] K. Alwasel, Y. Li, P. P. Jayaraman, S. Garg, R. N. Calheiros, and R. Ranjan, "Programming SDN-native big data applications: Research gap analysis," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 62–71, Sep. 2017.

[136] F. Xiao, J. Zhang, J. Huang, G. Gu, D. Wu, and P. Liu, "Unexpected data dependency creation and chaining: A new attack to SDN," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1512–1526.

[137] A. N. Alhaj and N. Dutta, "Analysis of security attacks in SDN network: A comprehensive survey," in *Contemporary Issues in Communication, Cloud and Big Data Analytics*, 2022, pp. 27–37.

[138] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 151–152.

[139] M. Brooks and B. Yang, "A man-in-the-middle attack against OpenDayLight SDN controller," in *Proc. 4th Annu. ACM Conf. Res. Inf. Technol. (RIIT)*, Sep. 2015, pp. 45–49.

[140] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, vol. 15, 2015, pp. 8–11.

[141] M. Wang, J. Liu, J. Chen, X. Liu, and J. Mao, "Perm-guard: Authenticating the validity of flow rules in software defined networking," *J. Signal Process. Syst.*, vol. 86, nos. 2–3, pp. 157–173, 2017.

[142] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[143] 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.

[144] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 165–166.

[145] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, "Software defined networking: Meeting carrier grade requirements," in *Proc. 18th IEEE Workshop Local Metrop. Area Netw. (LANMAN)*, Oct. 2011, pp. 1–6.

[146] R. Barrett, A. Facey, W. Nxumalo, J. Rogers, P. Vatcher, and M. St-Hilaire, "Dynamic traffic diversion in SDN: Testbed vs mininet," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 167–171.

[147] C.-C. Liu, Y. Chang, C.-W. Tseng, Y.-T. Yang, M.-S. Lai, and L.-D. Chou, "SVM-based classification mechanism and its application in SDN networks," in *Proc. 10th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jul. 2018, pp. 45–49.

[148] J. Xia, Z. Cai, G. Hu, and M. Xu, "An active defense solution for ARP spoofing in OpenFlow network," *Chin. J. Electron.*, vol. 28, no. 1, pp. 172–178, Jan. 2019.

[149] H. Y. Ibrahim, P. M. Ismael, A. A. Albabawat, and A. B. Al-Khalil, "A secure mechanism to prevent ARP spoofing and ARP broadcasting in SDN," in *Proc. Int. Conf. Comput. Sci. Softw. Eng. (CSASE)*, Apr. 2020, pp. 13–19.

[150] N. Ahuja, G. Singal, D. Mukhopadhyay, and A. Nehra, "Ascertain the efficient machine learning approach to detect different ARP attacks," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107757.

[151] S. Gao, Z. Li, B. Xiao, and G. Wei, "Security threats in the data plane of software-defined networks," *IEEE Netw.*, vol. 32, no. 4, pp. 108–113, Jul. 2018.

[152] Z. Liu, P. Longa, G. C. Pereira, O. Reparaz, and H. Seo, "Four$\mathbb{Q}$ on embedded devices with strong countermeasures against side-channel attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 3, pp. 536–549, May/Jun. 2018.

[153] D. Kreutz, F. M. V. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 55–60.

[154] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for OpenFlow applications," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 171–172.

[155] P. Pan and T. Nadeau, "Software driven networks problem statement," Netw. Work. Group Internet-Draft, Tech. Rep., vol. 15, Oct. 2011.

[156] M. Rahouti, K. Xiong, and Y. Xin, "Prototyping an SDN control framework for QoS guarantees," in *Proc. Int. Conf. Testbeds Res. Infrastruct.* Cham, Switzerland: Springer, 2020, pp. 3–16.

[157] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Participatory networking: An API for application control of SDNs," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 327–338, 2013.

[158] G. Yao, J. Bi, and L. Guo, "On the cascading failures of multi-controllers in software defined networks," in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2013, pp. 1–2.

[159] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *Proc. 23rd Int. Teletraffic Congr.*, 2011, pp. 1–7.

[160] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 413–424.

[161] J. Cao, Q. Li, R. Xie, K. Sun, G. Gu, M. Xu, and Y. Yang, "The crosspath attack: Disrupting the SDN control channel via shared links," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur.)*, 2019, pp. 19–36.

[162] X. Huang, K. Xue, Y. Xing, D. Hu, R. Li, and Q. Sun, "FSDM: Fast recovery saturation attack detection and mitigation framework in SDN," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Dec. 2020, pp. 329–337.

[163] K. Thimmaraju, L. Schiff, and S. Schmid, "Outsmarting network security with SDN teleportation," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Apr. 2017, pp. 563–578.

[164] D. Tatang, F. Quinkert, J. Frank, C. Röpke, and T. Holz, "SDN-guard: Protecting SDN controllers against SDN rootkits," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 297–302.

[165] S. Woo, S. Lee, J. Kim, and S. Shin, "RE-CHECKER: Towards secure RESTful service in software-defined networking," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–5.

[166] C. Röpke and T. Holz, "SDN rootkits: Subverting network operating systems of software-defined networks," in *Proc. Int. Symp. Recent Adv. Intrusion Detection*. Cham, Switzerland: Springer, 2015, pp. 339–356.

[167] E. Al-Shaer and S. Al-Haj, "FlowChecker: Configuration analysis and verification of federated openflow infrastructures," in *Proc. 3rd ACM Workshop Assurable Usable Secur. Configuration (SafeConfig)*, 2010, pp. 37–44.

[168] Q. Li, Y. Chen, P. P. C. Lee, M. Xu, and K. Ren, "Security policy violations in SDN data plane," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1715–1727, Aug. 2018.

[169] Z. Lu, F. Chen, G. Cheng, and S. Li, "The best defense strategy against session hijacking using security game in SDN," in *Proc. IEEE 19th Int. Conf. High Perform. Comput. Commun., IEEE 15th Int. Conf. Smart City, IEEE 3rd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2017, pp. 419–426.

[170] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[171] *Security-Enhanced Floodlight*. Accessed: Feb. 2022. [Online]. Available: http://www.sdncentral.com/education/towardsecure-sdn-control-layer/2013/10/

[172] M. P. Fernandez, "Comparing OpenFlow controller paradigms scalability: Reactive and proactive," in *Proc. IEEE 27th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2013, pp. 1009–1016.

[173] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–4.

[174] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed SDN controllers in a multi-domain environment," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–2.

[175] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 7–12.
[176] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Commun.*, vol. 11, no. 2, pp. 38–54, Feb. 2014.
[177] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, 2013, pp. 672–675.
[178] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE Local Comput. Netw. Conf. (LCN)*, Oct. 2010, pp. 408–415.
[179] S. Shirali-Shahreza and Y. Ganjali, "Efficient implementation of security applications in OpenFlow controller with FleXam," in *Proc. IEEE 21st Annu. Symp. High-Perform. Interconnects (HOTI)*, Aug. 2013, pp. 49–54.
[180] S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic clouds? Networks (or: How to provide security monitoring as a service in clouds?)," in *Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2012, pp. 1–6.
[181] R. Skowyra, S. Bahargam, and A. Bestavros, "Software-defined IDS for securing embedded mobile devices," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2013, pp. 1–7.
[182] P. Kazemian, G. Varghese, and N. McKeown, "Header space analysis: Static checking for networks," in *Proc. USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, vol. 12, 2012, pp. 113–126.
[183] P. Kazemian, M. Chan, H. Zeng, G. Varghese, N. McKeown, and S. Whyte, "Real time network policy checking using header space analysis," in *Proc. USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 99–111.
[184] S. Betgé-Brezetz, G.-B. Kamga, M. N. Balla, T. Criton, and H. Jebalia, "SDN-based trusted path in a multi-domain network," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (ICEW)*, Apr. 2016, pp. 19–24.
[185] J. H. Cox, R. J. Clark, and H. L. Owen, "A security policy transition framework for software-defined networks," in *Guide to Security in SDN and NFV*. Cham, Switzerland: Springer, 2017, pp. 149–169.
[186] A. Melis, D. Berardi, C. Contoli, F. Callegati, F. Esposito, and M. Prandini, "A policy checker approach for secure industrial SDN," in *Proc. 2nd Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2018, pp. 1–7.
[187] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 897–912, Apr. 2019.
[188] M. F. Hyder and M. A. Ismail, "Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches," *IEEE Access*, vol. 9, pp. 21881–21894, 2021.
[189] Q. Kang, L. Xue, A. Morrison, Y. Tang, A. Chen, and X. Luo, "Programmable in-network security for context-aware BYOD policies," in *Proc. 29th USENIX Secur. Symp. (USENIX Secur.)*, 2020, pp. 595–612.
[190] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Comput.*, vol. 10, no. 6, pp. 87–89, Nov. 2006.
[191] A. Voellmy and J. Wang, "Scalable software defined network controllers," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2012, pp. 289–290.
[192] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, nos. 1–3, pp. 1–6, May 1998.
[193] M. Belyaev and S. Gaivoronski, "Towards load balancing in SDN-networks during DDoS-attacks," in *Proc. 1st Int. Sci. Technol. Conf., Mod. Netw. Technol. (MoNeTeC)*, Oct. 2014, pp. 1–6.
[194] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: Detecting security attacks in software-defined networks," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2015.
[195] J. Amann and R. Sommer, "Providing dynamic control to passive network security monitoring," in *Proc. Int. Workshop Recent Adv. Intrusion Detection (RAID)*. Cham, Switzerland: Springer, 2015, pp. 133–152.
[196] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 310–317.
[197] T. Chin, X. Mountrouidou, X. Li, and K. Xiong, "An SDN-supported collaborative approach for DDoS flooding detection and containment," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2015, pp. 659–664.
[198] T. Chin, K. Xiong, and M. Rahouti, "SDN-based kernel modular countermeasure for intrusion detection," in *Proc. 13th EAI Int. Conf. Secur. Privacy Commun. Netw. (SecureComm)*. Cham, Switzerland: Springer, 2017, pp. 270–290.
[199] A. Khurshid, W. Zhou, M. Caesar, and P. B. Godfrey, "VeriFlow: Verifying network-wide invariants in real time," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 49–54.
[200] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 198–211, Jul./Aug. 2013.
[201] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *Proc. 9th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2013, pp. 18–25.
[202] D. Hock, S. Gebert, M. Hartmann, T. Zinner, and P. Tran-Gia, "POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.
[203] J. Ge, H. Shen, E. Yuepeng, Y. Wu, and J. You, "An OpenFlow-based dynamic path adjustment algorithm for multicast spanning trees," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (NDSS)*, Jul. 2013, pp. 1478–1483.
[204] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takács, and P. Sköldström, "Scalable fault management for OpenFlow," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 6606–6610.
[205] C. Monsanto, N. Foster, R. Harrison, and D. Walker, "A compiler and run-time system for network programming languages," *ACM SIGPLAN Notices*, vol. 47, no. 1, pp. 217–230, Jan. 2012.
[206] J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, A. R. Curtis, and S. Banerjee, "DevoFlow: Cost-effective flow management for high performance enterprise networks," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw. (Hotnets)*, 2010, p. 1.
[207] S. Wang, S. Chandrasekharan, K. Gomez, S. Kandeepan, A. Al-Hourani, M. R. Asghar, G. Russello, and P. Zanna, "SECOD: SDN sEcure control and data plane algorithm for detecting and defending against DoS attacks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–5.
[208] S. Wang, S. Chandrasekharan, K. G. Chavez, K. Sithamparanathan, A. Hourani, M. Asghar, G. Russello, and P. Zanna, "SECOD: SDN sEcure control and data plane algorithm for detecting and defending against DoS attacks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 448–452.
[209] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," in *Proc. 1st ACM Workshop Hot Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 121–126.
[210] A. K. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: Dynamic access control for enterprise networks," in *Proc. 1st ACM Workshop Res. Enterprise Netw. (WREN)*, 2009, pp. 11–18.
[211] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient OpenFlow-based networking," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2012, pp. 933–939.
[212] J. Martins, M. Ahmed, C. Raiciu, and F. Huici, "Enabling fast, dynamic network processing with clickOS," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 67–72.
[213] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 167–172.
[214] S. Kyung, W. Han, N. Tiwari, V. H. Dixit, L. Srinivas, Z. Zhao, A. Doupé, and G.-J. Ahn, "HoneyProxy: Design and implementation of next-generation honeynet via SDN," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 1–9.
[215] A. Lioy, G. Gardikis, B. Gaston, L. Jacquin, M. De Benedictis, Y. Angelopoulos, and C. Xylouris, "NFV-based network protection: The SHIELD approach," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–2.
[216] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software defined machine-to-machine communication for smart energy management," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 52–60, Oct. 2017.
[217] S. Scott-Hayward and T. Arumugam, "OFMTL-SEC: State-based security for software defined networks," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–7.

[218] T. Park, Y. Kim, V. Yegneswaran, P. Porras, Z. Xu, K. Park, and S. Shin, "DPX: Data-plane extensions for SDN security service instantiation," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*. Cham, Switzerland: Springer, 2019, pp. 415–437.

[219] T. OConnor, W. Enck, W. M. Petullo, and A. Verma, "PivotWall: SDN-based information flow control," in *Proc. Symp. SDN Res.*, Mar. 2018, pp. 1–14.

[220] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, "Rosemary: A robust, secure, and high-performance network operating system," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Nov. 2014, pp. 78–89.

[221] W. Han, H. Hu, Z. Zhao, A. Doupé, G.-J. Ahn, K.-C. Wang, and J. Deng, "State-aware network access management for software-defined networks," in *Proc. 21st ACM Symp. Access Control Models Technol.*, Jun. 2016, pp. 1–11.

[222] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.

[223] A. Shaghaghi, M. A. Kaafar, and S. Jha, "WedgeTail: An intrusion prevention system for the data plane of software defined networks," in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIACCS)*, Apr. 2017, pp. 849–861.

[224] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2013, pp. 1–16.

[225] R. Beckett, X. K. Zou, S. Zhang, S. Malik, J. Rexford, and D. Walker, "An assertion language for debugging SDN applications," in *Proc. 3rd ACM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2014, pp. 91–96.

[226] S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model checking invariant security properties in OpenFlow," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 1974–1979.

[227] M. Canini, D. Kostic, J. Rexford, and D. Venzano, "Automating the testing of OpenFlow applications," in *Proc. 1st Int. Workshop Rigorous Protocol Eng. (WRiPE)*, 2011, pp. 1–7.

[228] A. Leal, J. F. Botero, and E. Jacob, "Improving early attack detection in networks with sFlow and SDN," in *Proc. Workshop Eng. Appl.* Cham, Switzerland: Springer, 2018, pp. 323–335.

[229] N. Handigol, B. Heller, V. Jeyakumar, D. Maziéres, and N. McKeown, "Where is the debugger for my software-defined network?" in *Proc. 1st ACM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 55–60.

[230] A. Wundsam, D. Levin, S. Seetharaman, and A. Feldmann, "OFRewind: Enabling record and replay troubleshooting for networks," in *Proc. USENIX Annu. Tech. Conf.*, 2011, pp. 15–17.

[231] A. Sebbar, K. Zkik, Y. Baadi, M. Boulmalf, and M. D. E.-C. El Kettani, "Using advanced detection and prevention technique to mitigate threats in SDN architecture," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 90–95.

[232] Q. Shafi, A. Basit, S. Qaisar, A. Koay, and I. Welch, "Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network," *IEEE Access*, vol. 6, pp. 73713–73723, 2018.

[233] J. Malik, A. Akhunzada, I. Bibi, M. Imran, A. Musaddiq, and S. W. Kim, "Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN," *IEEE Access*, vol. 8, pp. 134695–134706, 2020.

[234] A. Ahmad, E. Harjula, M. Ylianttila, and I. Ahmad, "Evaluation of machine learning techniques for security in SDN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2020, pp. 1–6.

[235] P. P. Devi and B. Jaison, "Protection on wireless sensor network from clone attack using the SDN-enabled hybrid clone node detection mechanisms," *Comput. Commun.*, vol. 152, pp. 316–322, Feb. 2020.

[236] J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed, and S. A. Shah, "A time-efficient approach toward DDoS attack detection in IoT network using SDN," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3612–3630, Mar. 2022.

[237] X. Luo, Q. Yan, M. Wang, and W. Huang, "Using MTD and SDN-based honeypots to defend DDoS attacks in IoT," in *Proc. Comput., Commun. IoT Appl. (ComComAp)*, Oct. 2019, pp. 392–395.

[238] A. Nehra, M. Tripathi, M. S. Gaur, R. B. Battula, and C. Lal, "TILAK: A token-based prevention approach for topology discovery threats in SDN," *Int. J. Commun. Syst.*, vol. 32, no. 17, p. e3781, Nov. 2019.

[239] A. Sebbar, K. Zkik, Y. Baddi, M. Boulmalf, and M. D. E.-C. El Kettani, "Secure data sharing framework based on supervised machine learning detection system for future SDN-based networks," in *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*. Cham, Switzerland: Springer, 2021, pp. 355–371.

[240] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, "SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)," *Comput. Commun.*, vol. 160, pp. 697–705, Jul. 2020.

[241] E. Maccherani, M. Femminella, J. W. Lee, R. Francescangeli, J. Janak, G. Reali, and H. Schulzrinne, "Extending the NetServ autonomic management capabilities using OpenFlow," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2012, pp. 582–585.

[242] P. Varga, G. Kathareios, A. Máté, R. Clauberg, A. Anghel, P. Orosz, B. Nagy, T. Tóthfalusi, L. Kovács, and M. Gusat, "Real-time security services for SDN-based datacenters," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–9.

[243] L. Wang and D. Wu, "Seccontrol: Bridging the gap between security tools and SDN controllers," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Cham, Switzerland: Springer, 2017, pp. 11–31.

[244] Y. Park, N. V. Kengalahalli, and S.-Y. Chang, "Distributed security network functions against botnet attacks in software-defined networks," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–7.

[245] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2358–2372, Oct. 2018.

[246] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "Tennison: A distributed SDN framework for scalable network security," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2805–2818, Dec. 2018.

[247] B. E. Ujcich and W. H. Sanders, "Data protection intents for software-defined networking," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 271–275.

[248] M. B. de Freitas, P. Quitério, L. Rosa, T. Cruz, and P. Simoes, "SDN-assisted containerized security and monitoring components," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–5.

[249] M. B. D. Freitas, L. Rosa, T. Cruz, and P. Simões, "SDN-enabled virtual data diode," in *Computer Security*. Cham, Switzerland: Springer, 2018, pp. 102–118.

[250] N. M. Yungaicela-Naula, C. Vargas-Rosales, J. A. Pérez-Díaz, and M. Zareei, "Towards security automation in software defined networks," *Comput. Commun.*, vol. 183, pp. 64–82, Feb. 2022.

[251] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Towards a fully automated and optimized network security functions orchestration," in *Proc. 4th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2019, pp. 1–7.

[252] A. Hermosilla, A. M. Zarca, J. B. Bernabe, J. Ortiz, and A. Skarmeta, "Security orchestration and enforcement in NFV/SDN-aware UAV deployments," *IEEE Access*, vol. 8, pp. 131779–131795, 2020.

[253] F. Holik and P. Dolezel, "Industrial network protection by SDN-based IPS with AI," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Singapore: Springer, 2020, pp. 192–203.

[254] F. Valenza, S. Spinoso, and R. Sisto, "Formally specifying and checking policies and anomalies in service function chaining," *J. Netw. Comput. Appl.*, vol. 146, Nov. 2019, Art. no. 102419.

[255] A. M. Zarca, J. B. Bernabe, A. Skarmeta, and J. M. Alcaraz Calero, "Virtual IoT HoneyNets to mitigate cyberattacks in SDN/NFV-enabled IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1262–1277, Jun. 2020.

[256] P. R. Grammatikis *et al.*, "SDN-based resilient smart grid: The SDN-microSENSE architecture," *Digital*, vol. 1, no. 4, pp. 173–187, Sep. 2021.

[257] M. Repetto, D. Striccoli, G. Piro, A. Carrega, G. Boggia, and R. Bolla, "An autonomous cybersecurity framework for next-generation digital service chains," *J. Netw. Syst. Manage.*, vol. 29, no. 4, pp. 1–34, Oct. 2021.

[258] T. Chin, K. Xiong, and M. Rahouti, "Kernel-space intrusion detection using software-defined networking," *EAI Endorsed Trans. Secur. Saf.*, vol. 5, no. 15, p. e2, 2018.

[259] T. Dierks, "The transport layer security (TLS) protocol version 1.2," Tech. Rep. RFC5246, Aug. 2008.

[260] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: Threats, mitigations, and future directions," *J. Reliable Intell. Environ.*, pp. 1–39, Feb. 2022.

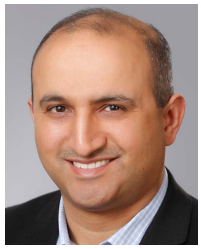
[261] C. Yoon, S. Lee, H. Kang, T. Park, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Flow wars: Systemizing the attack surface and defenses in software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3514–3530, Dec. 2017.
[262] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 239–250.
[263] M. Beshley, M. Klymash, H. Beshley, O. Urikova, and Y. Bobalo, "Future intent-based networking for QoE-driven business models," in *Future Intent-Based Networking*. Cham, Switzerland: Springer, 2022, pp. 1–18.
[264] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "BlockSDN: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Netw.*, vol. 34, no. 2, pp. 83–91, Mar. 2020.

MOHAMED RAHOUTI (Member, IEEE) received the M.S. and Ph.D. degrees in mathematics and electrical engineering from the University of South Florida, Tampa, FL, USA, in 2016 and 2020, respectively. He is currently an Assistant Professor with the Department of Computer and Information Sciences, Fordham University, Bronx, NY, USA. His current research interests include computer networking, blockchain technology, the Internet of Things (IoT), machine learning, and network security with applications to smart cities.

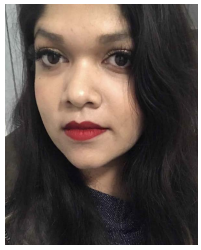
KAIQI XIONG (Senior Member, IEEE) received the Ph.D. degree in computer science from North Carolina State University. Before returning to academia, he was with IT industry for several years. He is currently a Professor with the Intelligent Computer Networking and Security Laboratory, University of South Florida, affiliated with the Florida Center for Cybersecurity, the Department of Mathematics and Statistics, and the Department of Electrical Engineering. His research was supported by the National Science Foundation (NSF), NSF/BBN, the Air Force Research Laboratory, Amazon AWS, the Florida Center for Cybersecurity, and the Office of Naval Research. His research interests include security, networking, and data analytics, with applications such as cyber-physical systems, cloud computing, sensor networks, and the Internet of Things. He received the Best Demo Award at the 22nd GENI Engineering Conference and the U.S. Ignite Application Summit with his team, in 2015. He also received the best paper award at several conferences.

YUFENG XIN (Member, IEEE) received the Ph.D. degree in operations research and computer science from North Carolina State University, Raleigh, NC, USA, in 2002. He is currently a Senior Researcher with the Renaissance Computing Institute, University of North Carolina at Chapel Hill. His research interests include networking, cloud computing, and cyber-physical systems.

SENTHIL KUMAR JAGATHEESAPERUMAL received the B.E. degree in electronics and communication engineering from Madurai Kamaraj University, Tamil Nadu, India, in 2003, and the Post Graduation degree in communication systems and the Ph.D. degree in embedded control systems and robotics from Anna University, Chennai, India, in 2005 and 2017, respectively. He is currently working as an Associate Professor with the Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India. He received two funded research projects from the National Instruments, USA, each worth USD 50,000 during the years 2015 and 2016. He also received another funded research project from IITM-RUTAG during 2017 worth Rs.3.97 Lakhs. His research interests include robotics, the Internet of Things, embedded systems, and wireless communications. He is a Life Member of IETE and ISTE.



MOUSSA AYYASH (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering. He is currently a Professor at the Department of Computing, Information, and Mathematical Sciences and Technology, Chicago State University, Chicago. He is also the Director of the Center of Information and Security Education and Research (CINSER). His current research interests include digital and data communication areas, wireless networking, visible light communications, network security, the Internet of Things, and interference mitigation. He is a member of the IEEE Computer and Communications Societies and the Association for Computing Machinery. He was a recipient of the 2018 Best Survey Paper Award from IEEE Communications Society.



MALIHA SHAHEED received the M.S. degree in cybersecurity from Fordham University, NY, USA, in 2021. She currently works as a Cybersecurity Consultant at Ernst and Young. Her current research interests include software defined networking security, the Internet of Things, and wearable authentication using soft biometrics.