International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 731-743
https://doi.org/10.70917/ijcisim-2026-5340
Article

Software-Defined Networking Security: Architecture, Attack Surface, and Resilient Mechanisms

Ajit Karki[1*], Vijay R[2*], Harishchandra A. Akarte[3], Abhijit Biswas[4]

[1]Department of Computer Science and Engineering, PhD Scholar, The ICFAI University, Tripura, Agartala, India
Faculty of Science and Technology, The ICFAI University, Sikkim, Gangtok, India
[2]Department of Computer Science, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Vel Nagar, Avadi, Chennai, India
[3]Department of Computer Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, Maharashtra, India
[4]Department of Computer Science and Engineering, The ICFAI University, Tripura, Agartala, India
Corresponding author's email id:ajitkarki@iusikkim.edu.in,vijayr@veltechmultitech.org

Abstract: Software-Defined Networking (SDN) is a new way of thinking about networking in which the networking function is split into two planes: control plane and data plane. The aim of SDN is to give network managers greater control over network configuration, increased programmability, flexibility, and efficient use of network resources. These features have driven the rapid-fire uptake of SDN in today's communications networks, cloud and data center. However, all these appealing features can create additional security problems, increasing the attack surface and putting critical elements at risk of breaches. With the rapid emergence of SDN, network security and resilience are emerging top topics of researchers' interests. This paper provides a comprehensive survey of the SDN security by covering its architecture, key benefits and potential vulnerabilities. It explores potential attacks on the data plane, attacks on the communication channels outside the control plane, security challenges in the control plane, and existing approaches and proposed countermeasures from the literature. Moreover, the paper presents a survey of existing work, lists open challenges, research gaps and future research directions and implements some new trends related to secure, scalable and resilient SDN environments.

Keywords: Software-Defined Networking (SDN), Network Security, SDN Architecture, Control Plane Security, Intrusion Detection Systems, OpenFlow Security, DDoS Attacks and Network Virtualization

1. Introduction

Recently, Software-Defined Networking (SDN) has soon transformed from a research idea to a commonly adopted networking paradigm; experimental and production SDN-enabled infrastructures are everywhere, creating a rapidly growing body of work that closely resembles areas such as virtualization, cloud, and the Internet of Things (IoT). SDN separates the control plane from the data plane, which allows network resources to be centrally managed, programmed and dynamically controlled. They are the concepts that were first introduced in architectures like Ethane and SANE, and have since been standardised as part of SDN, as it is a major enabler of cloud computing, network virtualisation and of next generation communication infrastructures.

SDN benefits have been proved in all types of networks: enterprise networks, data center networks, WANs, and carrier networks. For example, B4's network, deployed by Google, features all these attributes at a massive scale, helping to optimize traffic engineering, maximize the use of resources, and facilitate simpler use of the network through central control. Furthermore, SDN also provides substantial benefit in network visibility, monitoring, policy

enforcement, and traffic engineering; allowing administrators to finely tune network behaviours to meet changing requirements on the fly.

The SDN has both great potential and new security challenges. Its logically centralized controller allows for a global overview of the network, offering real-time monitoring, traffic analysis, and anomaly detection and quick reaction to security events. They have led to several security solutions such as intrusion detection systems, anomaly detection systems and dynamic policy enforcement as is the case with the FlowTags and SIMPLE. SDN is also programmable, allowing for automated security orchestration and fine-grained traffic control, further bolstering the security of today's networks.

The same features, which make the architecture attractive, also give rise to new attack surface areas, however. The SDN controller is a key element that can break the entire network when breached and is thus an appealing target for Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. Similarly, attack based on saturating the flow-table can be a problem for forwarding devices that have a small flow-table, degrading network performance significantly. In addition, SDN is open and programmable, which means it has issues with trust among the application, control, and data planes. Problems with OpenFlow-based implementations also complicate the security of SDN deployments, such as OpenFlow protocol vulnerabilities, policy verification, and security of controllers and software validation.

To tackle these, a wide array of different security mechanisms are proposed, including secure controller architectures, modular security services like FRESCO, security enforcement kernels, policy verification frameworks like VeriFlow and smart IDS. Though these methods have largely improved SDN security, in-depth security support from end to end using all SDN layers is still a research topic. Still, more research is needed on aspects such as controller resilience, secure communication channels, trust management, scalability, mitigating zero-day attacks and coordinating security across distributed controllers.

Driven by the efforts of these challenges, this paper proposes a thorough and up to date review of SDN Security. It investigates SDN architecture, its benefits, and the security threats posed by SDN, reviews the current defense techniques and mitigation strategies, and categorizes the existing research efforts systematically. Moreover, it identifies an existing research gap, presents the emerging trends in the field of software-defined networks (SDN) security, and proposes future research directions towards the design of secure SDN architectures that are scalable, resilient and intelligent.

1.1 The main contributions in this study are:

The results of this paper are derived by the following key contributions:

- Comprehensive analysis of SDN security issues in the light of SDN features and working: This paper involves comprehensive investigation of security issues of Software Defined Networking (SDN) with respect to SDN features and working.
- Layer Based Threat Classification: To gain a systematic view of security threats, categorize the different layers of the SDN they affect such as the application plane, control plane and data plane.
- It presents an extensive analysis of the current and emerging solutions -such as authentication, policy conflict resolution, controller replication, etc. and their capabilities.
- Research Gap Analysis: Research discovers its gaps in security needs and the demand for more effective, scalable and holistic security mechanisms.
- Implications for Emerging Technologies: It addresses the issues about the influence of SDN security challenges over new SDN technologies (Network Functions Virtualization (NFV) and design of algorithms), and the importance of increasing the security points in future SDN networks.

2. Literature review

Software-Defined Networking (SDN) has been one of the fastest growing technologies in the networking world and gaining much attention from academia and industry since it offers a centralized control, network programmability, flexibility and efficient management of resources. The control plane is separated from the data plane, which allows network administrators to configure and manage network behavior with software-driven policies, making network operations easier and more scalable. Hence, SDN has been broadly applied in cloud computing, data centres, Internet of Things (IoT) applications, wireless communication systems, edge computing and SDN in 5G/6G. Despite the various benefits that SDN has over the traditional networking architectures, the centralized control operation and programmable interfaces of the open networking create new security issues and point of vulnerabilities, which have become an area of foremost research.

In the last decade, there have been many studies conducted to explore different aspects of SDN design and implementation, such as SDN communication protocols, SDN security threats, SDN trust management, SDN access control, and SDN network monitoring techniques. Potential weaknesses at the application layer, control layer and infrastructure layer have been analysed, including denial-of-service (DoS) attacks, distributed denial-of-service (DDoS) attacks, attacks against the controller, flow-table overflow attacks, malicious application injection, spoofing attacks and man-in-the-middle attacks. These studies make it important to understand the trade-offs of centralizing network intelligence, which simplifies network management while introducing potential single points of failure and enticing targets for network attacks.

In the literature, many security mechanisms and countermeasures have been suggested to tackle the challenges. These include encryption methods like Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), and Transport Layer Security (TLS) for securing communication channels and protecting sensitive data. Further, RBAC based security, policy implementation system like FortNOX, IDS and IDP system, attack graph analysis, trust management models, and machine learning (ML) based approaches have been studied to improve the security of SDN. Recently, the integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques together with SDN controller has garnered more attention, which promises automated threat detection, anomaly identification, adaptive policy enforcement, and real-time attack mitigation.

Moreover, there have been various studies on the verification and policy enforcement of networks to guarantee correctness, consistency, and security in the network configurations. A number of directions have become significant areas of research for secure SDN deployment to guarantee SDN deployments reliability and trustworthiness, such as policy conflict detection, flow-rule verification, controller authentication and secure northbound and southbound interfaces. As SDN is more adopted in critical infrastructures, the creation of complete SDN security frameworks has become even more important as they can tackle both traditional and novel cyber threats.

We consider the literature survey on SDN architecture, threats, threat models, attack surface, security mechanisms, cryptographic security protection techniques, access control frameworks and policy verification approaches. The review will give a broad picture of the existing state of SDN security research and the challenges and opportunities faced by the dynamic of creating secure, scalable and resilient environment of SDN.

2.1 SDN nature and evolution

It can be traced back to the initial “SDN” research projects (Ethane, SANE) which led to the idea of SDN for centralised control of network policies enforcement [6] [5]. These papers were a breeding ground for what is currently known as SDN in determining that both the control logic as well as the forwarding devices could be separated. It was also during OpenFlow's announcement as a communication protocol standard that SDN became even more adoptable [9] and communication between control and data planes occurred.

With the further development, SDN can now be applied to various applications such network virtualization and cloud computing nowadays [3] [27]. The architecture has been very successful in a large-scale environment, such as the various traffic engineering and resource optimization in B4 network of Google for instance [1]. While SDN has

been successful at these, research has also uncovered a number of issues for SDN on a large scale including scalability, interoperability and reliability issues [2] [11]]. These ground works will then serve as the foundation for grasping SDN systems possibilities and restrictions.

2.2 Security threat landscape in SDN

Flexibility and programmability are some of the benefits SDN offers, but they also expand the attack surface of network infrastructures. The problem with it is it will have a (central) controller and that might be considered a single point of failure. Denial of Service (DoS) / Distributed Denial of Service (DDoS) attack can be used to attack the controller and as such, it affects the overall working of the network [17]. Moreover, the small flow table size of SDN switches can lead them to performance degradation and service failure due to attacks to exhaust their flow table [10].

Whilst the SDN is a programmable network, which is advantageous, it also includes the threats of trust and authentication between different layers of the network. OpenFlow installations have been identified with vulnerabilities in literature and a secure channel of communication and strong authentication mechanism need to be assured [7] [12]. Furthermore, traditional intrusion detection techniques can be simplified in SDN as it allows dynamically modifying the network topology and new adaptable and scalable network security solutions need to be designed [24].

A systematic approach to identify potential security vulnerabilities and threats from a SDN has been threat modelled by its developers by applying one of the threat modelling methodologies such as the STRIDE method [8]. These tests reveal that there are numerous attacks against SDN including Spoofing, Tampering and Information disclosure. This means that the threat landscape for the SDN is unique and requires an in-depth understanding to secure SDN.

2.3 Security mechanisms and defence structure

To address the listed vulnerabilities, several research studies have proposed security controls to SDN environment. One of SDN's drawbacks is that it enables flexible implementation of the policies using a centralized monitoring and dynamic enforcement system. One such example is CloudWatcher system that leverages the SDN capability for network-wide security monitoring, and anomaly detection [19]. The same applies to traffic anomaly detection in which the study results indicated that SDN is proven a legitimate tool to detect any malicious activity in real time [24]. Fine-grained network traffic access control between the dynamically deployed security policies in various network devices is accomplished in frameworks such as FlowTags and SIMPLE [14] [15]. This kind of policy enhances the likelihood of consistency, and compliance with security needs. In addition, with techniques like OpenFlow-based random host mutation, moving target defense (MTD) can be achieved by an attacker must be able to breach a fortified defense network with stationary network settings [16]. Other articles have discussed the possibilities of applying SDN to intrusion detection and prevention systems (IDS/IPS) to real-time detect threat status to mitigate the threat based on programmable network controls [22] and [23]. The following solutions point to the promise SDN can bring to truly transform how networks are secured by amalgamating the elements of monitoring, analysis and response under a single (solution) roof.

2.4 Modifications in Policy & Planning

This concern over SDN correctness and the settings' security becomes a significant concern, in a network context that is dynamic and programmable. Many tools and frameworks have been implemented to address this issue. One of them is VeriFlow, which assists verifying properties across the network, which are considered to be “invariants” in real-time with the aim of ensuring that network security policies are being followed across the network [29]. Likewise, FlowChecker is able to provide configuration analysis and verification for federated SDN networks [26]. To perform model checking the different security properties using model-checking techniques [28] have also been applied to determine whether SDN breaks any of the properties prior to deployment. These are proper steps to

avoid any security compromises, which only can occur due to improper configurations. Furthermore, it is possible to test and validate SDN apps based on tools such as NICE, to guarantee their correctness and reliability [25]. In addition to the verification, attempts have been made to develop extensible security frameworks, which are modular. An example of the same is FRESCO, which offers a tool to create security services in SDN infrastructures, which allow different forms of protection to be used [32]. In the same vein, security enforcement kernels have been suggested to offer a centralized point of control when it comes to enforcing network policies [33]. Such solutions can help augment and extend the flexibility of SDN security solutions.

2.5 Research gaps and challenges

While a lot has been accomplished so far, there are several more challenges to overcome in terms of SDN security. Most of the solutions focus on some aspects of security, such as intrusion detection or policy enforcement and are unable to provide a framework that can be used on all levels of the SDN architecture. Thus, the issue of how to manage the trust of the app plane, control plane, and data plane are topics ripe for research, especially in multi-tenant and distributed environments [7]. The centralized controller is also an interesting feature since it could be a potential traffic congestion issue in a large-scale network implementation. While distributed controller architectures are proposed, these add to the added complexity needed in providing consistency and security among multiple controllers [11]. Furthermore, new technologies, such as Internet of Things (IoT) and edge computing and SDN generate new challenges to delve into. In sum, this literature seems to stress the urgency to adopt a holistic strategy to SDN security threat, using a mix of cryptography, live monitoring and smart implementation of security policies.

Through the literature search, it is proved that SDN provides many advantages namely flexibility of services, programmability of the services and centralized control. The advantages, however, are associated with security dilemmas, which need to be handled with utmost care. Many proposals have been made but up to date there are no comprehensive and comprehensive security frameworks. This motivates further research to provide more security mechanisms like hybrid cryptographic schemes, smart threat detection models and so on that could make SDN deployable in the existing network architectures.

3. Security analysis of SDN

Confidentiality, Integrity, Availability, Authentication and Non-Repudiation are well known principles of secure communication networks [4]. It is crucial to provide protection to these properties, which entails protecting both the data and the network resources, including forwarding devices and other control entities, and communication processes against malicious attack and accidental disruption. In fact, this Software-Defined Networking (SDN) concept, which alters the whole perspective of networks in traditional setups, requires rethinking of its existing security models, and care must be taken to ensure that the existing fundamental principles are maintained. The architectural paradigm launched by SDN with the separation of the control plane and the data plane and the logical centralization of the control mode brings with it opportunities and challenges on the security front. Casado et al. [5] who focused on the security implications of a split control and forwarding architecture, and who designed the architecture of “Secure Architecture of Networked Enterprises (SANE) have performed one of the first studies on such architectures. This method employed logically centralized controller system that provides access control policy and manages authentication of the host in it so that access control can be performed at a fine-grained level and visibility is provided across network. Irrespective of its conceptual advantages, the SANE architecture was initially seen as a limiting solution because of its demand of significant changes to current network infrastructures and end-host settings. These requirements posed a challenge in enterprise environment as far as scalability and deployability were concerned. However, this initial effort played a major role in the development of SDN as it pointed out the trade-offs between centralized control and security and hence became the foundation of future work in SDN security architectures.

The Ethane architecture [6] expanded on the original concepts of SANE, and presented a more realistic and viable version, where the changes were fewer to the existing network architectures. Ethane employed specialized switches that used to relay packets from rules written on flow tables and a logically centralized controller applied global network policies. This separation of the control and data plane components of networking caused control and

requirements to be more flexible yet considerably more programmable, and this represented a giant leap towards implementation of modern SDN and OpenFlow based system. Useful as the Ethane architecture was, it had a number of drawbacks that previewed security concerns that modern SDN deployments pose. Among the key issues is the vulnerability of network policies to attack by application-layer vulnerabilities. One major aspect of SDN networks that are important today is the role of an application that supports services such as Network Functions Virtualization (NFV) and runs on top of the SDN controller. As a result, application intrusions may spread to the rest of the network compromising policy enforcement and overall system integrity. These risks can be systematically evaluated during the architectural exploration of SDN stack, including application plane, control plane, data plane, and the interfaces which glue them together. Applying the context of the security concerns with SDN to the SDN framework (Fig. 1), we can outline the challenges related to each layer of the framework: application, control and data planes, and the interfaces between the planes.

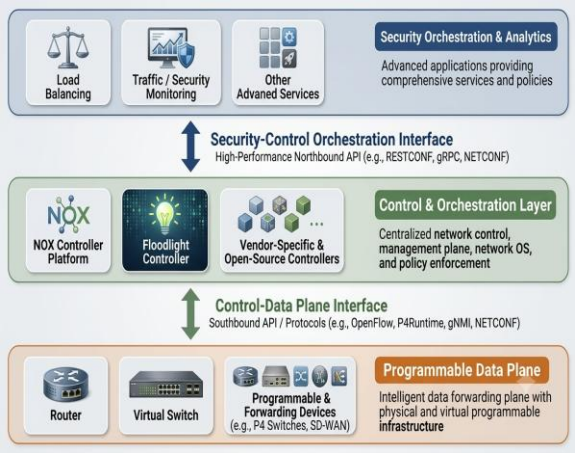

Figure. 1 SDN Functional Architecture illustrating the data, control and application layers and interfaces

However, there are new vulnerabilities associated with SDN thanks to the changes to the architecture and interactions made by various components that have been identified by ongoing security research. One example can be the detailed analysis of potential vulnerabilities of the OpenFlow protocol, applying a threat-modelling framework such as STRIDE (Denial-of-Service (DoS), Information disclosure etc.), which were only suggested, but no attempt was made to validate the usefulness of this threats in real-life situations, so further questions are left regarding applicability of the mitigation proposals in practical use. In the case of OpenFlow specification, a possibility is to establish communication channels between the controllers and the switches by means of TLS, which does not mention any mutual authentication; and is considered optional [9]. However, this level of security doesn't need to be done in the protocol. After vulnerability tests it has been noticed that in majority of the commercial implementations, the TLS has not been used to protect the network, which puts it at risk of injective and changeable rules that can affect the integrity of the network [10]. Based on Kreutz et al. [11] a broader perspective on SDN Security provides insights of the emerging threats, which are target of SDN Security: centralized control plane and network programmability. They provide helpful tips as to aspects that might be relaxed to make a more resilient system - such as controller replication, software splitting and safe parts etc. Furthermore, certain attack vectors can be detected through experiments deployed in ProtoGENI testbed - the propagation of malicious traffic, the wormhole between users and the flooding attack via massive scale attack [12].

As mentioned above, each study on the different, but yet similar, security concerns of SDN type of architecture has always been constant. Since there are a lot of these vulnerabilities, a systematic discussion by type of attack (e.g. unauthorized access DoS etc.) should be made, accompanied with a layer/interface at which an attack happens (SDN in particular is highly vulnerable to these factors). Mainly because there are no such central control unit, (central control unit is called SDN-Controller) and programmable data path unit (Data Plane Unit) changes in the SDN network graphics when compared with traditional network graphics. Although these vulnerabilities exist in the literature to

date have focused on how SDN can be used to provide a more secure network rather than investigating the vulnerabilities. However, due to this inequity, people have started questioning whether a solution more specific and detailed (in the control plane or data plane), is required to solve the SDN infrastructures problem. The above problems related to SDN and network security will be presented and SDN will be discussed how it is going to solve above problems.

4. Security enhance-SDN

Controls software: can be controlled by software defined network (SDN) enabling some new controls to network. When the global or network wide view is correlated with network programmability helps in the Intelligence collection process based on current Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) and then analyse them and centrally re-program the network. In contrast to traditional networks, it may also make the SDN network more resilient towards coping with malicious attacks.

Table 1. Definition of terms

Security Issue / Attack Category	**Example**	**Application Layer**	**App-Control Interface**	**Control Layer**	**Control-Data Interface**	**Data Layer**
Unauthorized Access	Unauthorized controller access	✓	✓	✓		
	Unauthenticated application	✓	✓	✓		
Data Leakage	Flow rule discovery (side-channel attack on input buffer)					✓
	Forwarding policy discovery (packet processing timing analysis)					✓
Data Modification	Flow rule modification affecting packet forwarding		✓	✓	✓	
Malicious Applications / Control Attacks	Fraudulent rule insertion	✓	✓	✓		
	Controller hijacking	✓	✓	✓		
Denial of Service (DoS)	Controller-switch communication flooding		✓	✓	✓	

	Switch flow table flooding					✓
Configuration & Design Issues	Lack of TLS (or weak authentication mechanisms)		✓	✓	✓	
	Policy enforcement inconsistency	✓	✓	✓		

5. Security challenges in SDN

Wherever “While security is a major positive for Software-Defined Networking, not enough has been done to address the practical issues associated with SDN systems' security. Because SDN can be programmed, changing the behavior of the network can be done dynamically and flow based policies can be easily applied. However, this versatility can also pose potential security threats, particularly in the event that policies are entered and updated dynamically in numerous gadgets and applications. Therefore, it is highly advised to be aware of the correctness and consistency of network security policies. For achieving this, formal verification methods such as model checking are increasingly important components in identifying inconsistencies and conflict in policies. For instance, the combination of model checking and symbolic execution has been employed for correctness checking of OpenFlow apps, and, binary decision diagrams have been applied to detect misconfigurations of the flow tables of switches. This is serious -it can help ensure that SDN systems are not just secure, but functioning as they are supposed to be. In SDN, policies can be dynamically generated and changed, resulting in inconsistencies and conflicts. This occurs when policies differ on various devices and/or applications, or the polices change often. To prevent this, formal verification methods may be employed to verify them and ensure they are correct and consistent. This can be accomplished with model checking, the idea being to build up a model of the system and test this model against a set of properties/rules. To improve the security and reliability of SDN systems formal verification techniques can be used. This is particularly crucial in networks where security plays a crucial role, like in monetary institutions or even authorities networks. However, it is equally important that, in the interest of security, there are some issues that need to be addressed in the case of SDN technology, such as formal verification techniques. In conclusion, making SDN infrastructures secure is a complicated undertaking, and there are certainly hazards and challenges that must be taken under consideration. However, SDN systems can be made more secure and reliable with formal verification and other security measures to ensure the widespread adoption of SDN systems.

A tool known as FlowVisor was developed to make better separation of networks and the enforcement of rules. It pioneered the concept of ‘network slicing' that segment network resources into different, isolated slices which do not touch each other. Some other tools, such as FlowChecker and Flover, are based on the notion to test whether the network configuration and set up is accessible or not, and to verify validity of the rules used for controlling flows. There is another tool, VeriFlow that tests network rules as they are executed by placing an additional layer between the network controller and network devices handling the data transfer. This layer filters out any rules that are incorrect before they are applied. VeriFlow has a fast rule checking speed, but it is unable to deal with large and complex network systems that have a large number of controllers.

Apart from verification-based solutions, some language-based security approaches have been suggested to enable a safe programming of networks and their enforcement of security policies. For instance, language based-approaches allow for such a fine-grained access control by incorporating external authentication solutions in SDN applications [30]. More recently, the idea of “splendour eizolatos” has been put forward to ensure traffic separation, with the aim of programming-based confidentiality and integrity enforcement [31]. These efforts point us towards an increased emphasis on the research community to address policy conflicts and secure policy deployment in SDN environment.

Over the past few years we have come a long, long way when it comes to software-defined networking (SDN). Still, however, despite all this progress there is a significant gap in the matter of having a comprehensive framework and guidelines for designing or implementing secure SDN system. One that is attempting to do this is FRESCO, a framework for the development of security applications for OpenFlow. Those threats can easily be detected and mitigated as part of the library of reusable components included in FRESCO. It has a mechanism named "FortNox enforcement kernel" that helps guarantee security regulations are non-contradictory, uniform and complete. This is significant because it allows security personnel to rapidly and conveniently install new security services, with no need to start from scratch. While FRESCO is an important step forward, it is not a complete solution - we still need a more unified approach to SDN security if we are going to really protect our networks. Operational issues such as auditing and compliance, as well as network visibility are always at the forefront of the conversation. The accurate inventory of the network devices, configurations and interconnections required to effectively manage a network is an increasingly complex challenge in virtualized and dynamic SDN networks. Additionally, virtualization and mobility are further complicating the mapping and verification of network states. The research reported in this paper aims at tackling these problems by investigating for advanced verification and debugging techniques. For example, the scalability and security issues of OpenFlow-based systems on cyber-physical environments have been studied [34] and tools, like network debuggers, have been suggested to track the execution path to determine the root causes of an anomaly in SDN applications [35].

Table 2. Categorization of Research on Security in SDN

Research Work	Security Analysis	Enhancement	Solution	Application Layer	App-Control Interface	Control Layer	Control-Data Interface	Data Layer
[7], [10], [12]	✓			✓	✓	✓	✓	✓
[11]	✓			✓	✓	✓	✓	✓
[5]				✓	✓	✓		
[13], [14], [21]		✓	✓	✓	✓	✓	✓	✓
[15]		✓		✓	✓	✓	✓	
[16]		✓		✓	✓	✓	✓	✓
[17], [24]		✓		✓	✓	✓	✓	✓
[18], [19]		✓		✓	✓	✓	✓	✓
[20], [22]		✓		✓	✓	✓	✓	✓
[23]			✓	✓	✓	✓		
[25]			✓	✓	✓	✓	✓	✓
[26], [28] [30], [32]			✓	✓	✓	✓	✓	✓
[31]			✓	✓	✓	✓		
[33], [34]			✓	✓	✓	✓	✓	✓
[35]			✓	✓	✓	✓	✓	

When discussing the architecture of Software-Defined Networking (SDN), it is useful to envision it as being separated into layers, with layers being the application layer, control layer and data layer with well-defined interfaces

among each layer. However, as we have witnessed previously, every one of these layers and interfaces can boast security susceptibilities. Therefore, researchers have been developing solutions that focus on specific layers/interface. Some early design, such as SANE, have provided us with valuable insights as to how to think about security on each of the layers, but are not complete solutions themselves. Rather, they mark set of ideas that have influenced the notions of security for SDN. The conclusion here is that the overall research effort is currently a bit piecemeal, and that what is needed are multi-layered comprehensive security solutions able to cope with the varying SDN challenges.

6. Discussion

The majority of security research that has been performed on networks advantages Software-Defined Networking (SDN) to increase network security but overall does not actually go anywhere near addressing the security concerns that SDN poses. Much research effort has been devoted to integrating SDN with a middlebox (via different methods) or network monitoring system (e.g., Open Network Summary) in order to provide security services or detect bad traffic in real-time. During the operation of networks, this can be useful in support of their security but is only a sort of a band-aid. Such solutions are not necessarily effective when it comes to addressing the underlying SDN security issues. For instance, researchers have been attempting to use SDN to improve the networks' ability to identify and prevent malicious traffic, but they are not necessarily considering how SDN could make networks vulnerable. Therefore, if the networks are going to be made secure, it is important to understand and correct these underlying issues. That has not to say that we cannot make SDN more secure, however, as well as bolting on security, there is also a need to think carefully about the way it is designed so security is built in. In this way, networks can be developed that are more secure, more dependable and capable of fending off a variety of threats. It is particularly necessary with networks becoming increasingly vital to everyday life and risks of cyber-attacks skyrocketing and other security issues. So, while current research on SDN and network security is a step in the right direction more studies are needed to really find the answers to why SDN is vulnerable in terms of network security.

Alternate approaches to network protection, such as "moving target defense", are just one example of how software defined networking (SDN) can be programmed and modified to remain secure. These techniques involve altering the network configuration such as variable IP addresses, to make it difficult for the attacker to obtain information and make an effective attack. If the network is continually evolving then it makes it more difficult for attackers to calculate and execute attacks, improving the overall resiliency of the network. However, all these approaches impose requirements on the SDN control systems themselves when they come to security, as if they are not secure enough, they can be compromised.

Based on the work that has been done to date, it is obvious that the majority of companies are building software-defined networks with OpenFlow to integrate the control plane and data plane. Nevertheless, although OpenFlow has been key to helping SDN get going, do not get the idea that it is the sole means of achieving it. Other protocols and frameworks exist such as LISP, NETCONF, ForCES, PCE and I2RS, which could function in a comparable way, but with differing security concerns. As well, enterprises are developing their own mobile security policies, thereby complicating security issues. Therefore, we should be mindful of vulnerabilities in OpenFlow when we apply them to these other interfaces-security issues at the edges of the control and data planes and between applications and the control plane are likely to be experienced. All these various protocols need to be taken into account and what their impact might be on security. Therefore, it is not only OpenFlow, it is the other ways for control and data planes to communicate with each other. Therefore, as companies are devising their own protocols, it is increasingly complex. It is important to ensure we have an understanding on how all these different protocols are run, and where they could be attacked. In understanding the security issues of OpenFlow, it may be possible to gain more insight into how to address similar issues for other protocols. However, it is not the universal remedy; careful consideration needs to be given to how what has been learnt can be used in each particular circumstance. Need to keep abreast of any new developments, of course, as security situations never remain the same. The bottom line, then, is that everyone should learn about all of the various protocols in general, and how they might affect security. We cannot just focus on OpenFlow - we have to think about the bigger picture, and how all these different pieces fit together. Therefore, when

we do that, we will have some more complete information about the security issues we are in, and how to address them.

After reading the headlines, we are not left with the impression that we are seeing as much work being done to find security solutions as we see to analyse security problems. Much research has been conducted to identify vulnerabilities and these have largely been contained in the control plane and data planes, but are only concerned with particular issues. Some, for instance, only consider how to solve the policy conflicts arising between the application and control interface. This represents a significant research gap, as it would be ideal to have solutions that could deal with a variety of security challenges simultaneously. There are many potential threats at every layer of SDN; clearly, there is need for more holistic and integrated security solutions. It is important to come up with strategies which will take care of all the various security threats rather than tackling each one individually. This will enable us to develop powerful and effective security solution that can resist against multiple threat.

In Software-Defined Networking (SDN) these days, it is evident that security is a major concern. Various efforts are being undertaken to solve those security issues, such as the Open Networking Foundation and the European Telecommunications Standards Institute (ETSI). They have special teams devoted to creating SDN and Network Functions Virtualization (NFV) more secure. Other organizations (including International Telecommunication Union and Internet Research Task Force) are performing research into SDN security as well. They all share the concept that security must be taken into account from the outset of designing SDN systems and not added onto. This will mitigate a security difficulty and ingest a good deal SDN a lot more reliable. Starting with security in mind can help create more robust and more trusted networks.

Network security cannot be taken lightly, particularly related to new technology offerings, such as Software-Defined Networking (SDN). Although SDN is being deployed in various locations, SDN security has not been adequately considered to protect against large attacks. This is not a good thing particularly of a large network covering a large area. Developing new strategies, guidelines and policies to proactively and organized address SDN security issues. By concentrating on security during this early SDN evolution stage, problems such as poor performance, limited functionality and vulnerabilities to future networks can be avoided. In so doing, we can ensure that SDN is safe and reliable - a key condition for its success.

7. Conclusion and future work

This paper has discussed in detail the Security Challenges in Software-Defined Networking (SDN) to show its potential benefits as well as inherent security vulnerabilities. The study reveals that there is a double-edged aspect of SDN in terms of security. But on the other, since it is centralized and its traffic is easily visible globally, the centralized features make it incredibly effective for executing advanced security solutions such as real-time threat monitoring, dynamic policy enforcement, and swift threat remediation. However, these capabilities also create new avenues for attack: attacks focusing on the controller, flow rule manipulation, and attacks present due to programmable interfaces.

A systematic classification of the SDN security issues, based on the architectural layers and attack surfaces, has been used to critically review the past research efforts, and to determine the key limitations and shortcomings of existing attacker solution sets. The findings show that there is a great disparity between how security and defence are evolving and the ability of the milder measures currently proposed to counter them, with many of the proposed measures treating individual aspects of security in isolation rather than offering a more comprehensive approach. This means that even with the advances made, still, it is an open challenge to build a completely secure and resilient SDN environment.

To elevate, it is necessary to make systems with many layers and systems that are strong. These systems need to be smartly able to identify threats, and automatically implement policies. This can be achieved with a combination of several types of encryption, such as a combination of symmetric and asymmetric. This can assist protect interaction between various areas of a network to make sure safety and security. These hybrid encryption methods can help us ensure our networks are more secure, and protected against potential cyber-attacks. It is particularly relevant to

software-defined networks (SDN) where security is a critical issue as it could jeopardize data integrity and security by allowing breaches.

Some big issues remain needing to be addressed: Some of the issues that need to be addressed are the failures of the app-level deployment to ensure it is secure; controller scalability; and trust management issues. Nevertheless, more so, if we look at SDN and new technologies such as NFV, IoT and edge computing, it brings new security concerns, which have to be studied in detail.

To maximise the utilisation of SDN in the next generation of networks, an optimum balance needs to be achieved. This will involve exploiting its programmability and its intelligence, but will also have security-related issues that we must address. New and policy-scale solutions need to be developed to reduce these risks. In so doing, we will gain the full benefit of SDN, and enjoy more efficient and secure networks.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Ajit Karki did the concept and design, data collection and data analysis and interpretation. Vijay R assisted with analysis and interpretation of data and made critical revisions to the manuscript. Harishchandra A. Akarte contributed to the development of the educational media and manuscript preparation. Abhijit Biswas assisted with intervention implementation and manuscript revision. The authors have read and accepted the final manuscript and do take responsibility for the entire content of the paper including integrity and accuracy.

Acknowledgments

The authors would like to acknowledge the support and guidance received during the course of this research.

REFERENCES

1. S. Jain et al., "B4: Experience with a globally-deployed software defined WAN," in Proc. ACM SIGCOMM, 2013, pp. 3-14.
2. S. Sezer et al., "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Commun. Mag., vol. 51, no. 7, pp. 36-43, Jul. 2013.
3. ETSI, "Network Functions Virtualization (NFV): Introductory White Paper," 2012. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper.pdf
4. C. Douligeris and D. N. Serpanos, Network Security: Current Status and Future Directions. Hoboken, NJ, USA: Wiley, 2007.
5. M. Casado et al., "SANE: A protection architecture for enterprise networks," in Proc. USENIX Security Symp., 2006.
6. M. Casado et al., "Ethane: Taking control of the enterprise," ACM SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 1-12, Oct. 2007.
7. R. Kloeti, "OpenFlow: A security analysis," Master's thesis, ETH Zurich, 2013. [Online]. Available: ftp://yosemite.ee.ethz.ch/pub/students/2012-HS/MA-2012-20_signed.pdf
8. S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling using the STRIDE approach," MSDN Mag., pp. 68-75, 2006.
9. Open Networking Foundation, "OpenFlow Switch Specification Version 1.3.2." [Online]. Available: https://www.opennetworking.org
10. K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in Proc. ACM HotSDN, 2013, pp. 151-152.
11. D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proc. ACM HotSDN, 2013, pp. 55-60.
12. D. Li, X. Hong, and J. Bowman, "Evaluation of security vulnerabilities using ProtoGENI as a launchpad," in Proc. IEEE GLOBECOM, 2011, pp. 1-6.
13. B. Anwer et al., "A slick control plane for network middleboxes," in Proc. Open Networking Summit, 2013.
14. S. Fayazbakhsh et al., "FlowTags: Enforcing network-wide policies in the presence of dynamic middlebox actions," in Proc. ACM HotSDN, 2013.
15. Z. A. Qazi et al., "SIMPLE-fying middlebox policy enforcement using SDN," in Proc. ACM SIGCOMM, 2013.
16. J. H. Jafarian, E. Al-Shaer, and Q. Duan, "OpenFlow random host mutation: Transparent moving target defense using software-defined networking," in Proc. ACM HotSDN, 2012, pp. 127-132.

17. R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proc. IEEE LCN, 2010, pp. 408-415.
18. J. R. Ballard, I. Rae, and A. Akella, "Extensible and scalable network monitoring using OpenSAFE," in Proc. INM/WREN, 2010.
19. S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks," in Proc. IEEE ICNP, 2012, pp. 1-6.
20. A. K. Nayak et al., "Resonance: Dynamic access control for enterprise networks," in Proc. ACM WREN, 2009, pp. 11-18.
21. J. Naous et al., "Delegating network security with more information," in Proc. ACM WREN, 2009, pp. 19-26.
22. R. Skowyra et al., "Software-defined IDS for securing embedded mobile devices," Boston Univ., Tech. Rep., 2013.
23. A. Goodney et al., "Pattern-based packet filtering using NetFPGA in DETER infrastructure," 2013. [Online]. Available: http://fif.kr/AsiaNetFPGAws/paper/2-2.pdf
24. S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software-defined networking," in Recent Advances in Intrusion Detection. Berlin, Germany: Springer, 2011, pp. 161-180.
25. S. R. Islam, M. M. Rahman, M. A. Rahman, and M. S. Hossain, "A comprehensive survey of vulnerability and information security in SDN," *Computer Networks*, vol. 206, Art. no. 108802, Apr. 2022, doi: 10.1016/j.comnet.2022.108802.
26. M. A. A. Aladaileh and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors*, vol. 23, no. 9, Art. no. 4441, 2023, doi: 10.3390/s23094441.
27. T. E. Ali, Y.-W. Chong, and S. Manickam, "Comparison of ML/DL approaches for detecting DDoS attacks in SDN," *Applied Sciences*, vol. 13, no. 5, Art. no. 3033, 2023, doi: 10.3390/app13053033.
28. Z. Liu, Y. Wang, and F. Feng, "A DDoS detection method based on feature engineering and machine learning in software-defined networks," *Sensors*, vol. 23, no. 13, Art. no. 6176, 2023, doi: 10.3390/s23136176.
29. B. Ayodele and V. Buttigieg, "SDN as a defence mechanism: A comprehensive survey," *International Journal of Information Security*, vol. 23, pp. 141-185, 2024, doi: 10.1007/s10207-023-00764-1.
30. D. Sendil Vadivu and N. Rajagopalan, "Software-defined network planes-A survey on attacks and countermeasure," *International Journal of Communication Networks and Distributed Systems*, vol. 29, no. 6, pp. 598-630, 2023, doi: 10.1504/IJCNDS.2023.133900.
31. M. R. Alauthman *et al.*, "Towards a machine learning-based framework for DDoS attack detection in software-defined IoT (SD-IoT) networks," *Engineering Applications of Artificial Intelligence*, vol. 123, Art. no. 106432, Aug. 2023, doi: 10.1016/j.engappai.2023.106432.
32. M. A. Alkasassbeh *et al.*, "Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks," *Computers & Security*, vol. 134, Art. no. 103462, Nov. 2023, doi: 10.1016/j.cose.2023.103462.
33. "Deep-discovery: Anomaly discovery in software-defined networks using artificial neural networks," *Computers & Security*, vol. 132, Art. no. 103320, Sep. 2023, doi: 10.1016/j.cose.2023.103320.
34. "A comprehensive survey of distributed denial of service detection and mitigation technologies in software-defined network," *Electronics*, vol. 13, no. 4, Art. no. 807, 2024.
35. M. A. Albahar, A. S. Alqahtani, and M. A. Alshahrani, "A comprehensive survey on software-defined networking security: Threats, vulnerabilities, and countermeasures," Journal of Network andComputer Applications, vol. 214, Art. no. 103615, 2023.