Survey of Security in Software-Defined Network

Nadya El Moussaid(✉), Ahmed Toumanari, and Maryam El Azhari

LISTI, National School of Applied Sciences,
Ibn Zohr University, Agadir, Morocco
nadya.elmoussaid@edu.uiz.ac.ma, atoumanari@yahoo.fr,
maryam.ensa@gmail.com

Abstract. The requirements of cloud computing are putting the traditional networks in tension which influence the quality of the services provided by cloud computing. Therefore, the application of software defined-network (SDN) within cloud computing reinforces the dynamicity and flexibility of cloud. Recently, SDN is the trend in networking and virtualized networks, where, SDN separate the network control plane from the data plane, which leads the management of the network routing from decentered architecture to centered architecture. Despite the advantages of merging the SDN paradigm within the cloud environment, the security issues still in the surface. This paper presents a survey on the security issues in software-defined networking and the challenges faced by admins and providers in order to guarantee a secure environment with a resume about the proposed solution.

Keywords: Software-defined network · SDN · SDN security · Security · Privacy

1 Introduction

The cloud computing introduced the unlimited virtualized resources that changed the way of accessing and storing data. The cloud characterized with the five essential characteristics namely: (1) Resource pooling, (2) On-demand capabilities, (3) Broad network access, (4) Rapid elasticity and (5) Measured services. The providers offer these characteristics in the form of three major services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Because of this attractiveness of the cloud, several organizations migrate from adopting traditional warehouse infrastructure to utilizing services provided by cloud computing [1]. Also, attracts the attackers to seeking for any vulnerability that can help them getting access to sensitive data or to get benefit of the advantages of the cloud in order to exercise attacks from the cloud against other organization.

The virtualization and the shared resource between multiple tenants are the backbones of cloud computing, the virtualization can be as a virtual machine and a virtual network. Software-defined networking is part of the virtualization systems, the use of SDN technology may improve the performance of network routing within the cloud

M. Ezziyyani et al. (eds.), *Advanced Information Technology, Services and Systems*,
Lecture Notes in Networks and Systems 25, https://doi.org/10.1007/978-3-319-69137-4_21

computing. However, it increases the sensitivity to security issues namely confidentiality, integrity and availability issues.

The SDN is an element of the software-defined system (SDS) package that contains:

- Software-defined Networking (SDN)
- Software-defined Cloud Networking (SDCN)
- Software-defined Storage (SDS)
- Software-defined Data Center (SDDC)
- Software-defined Radio (SDR)

SDN provides five major benefits [2], that we quote:

- Accuracy: The IT resources become automatic and programmable. Also, the requests of clients are independent of the hardware.
- Agility: The agility enables the components from migrating between environments in an easy and flexible way.
- Adaptability: this property provides no reliance on hardware resources of the vendors, which leads to the adaptability to new configurations and environments.
- Assurance: SDN provides an assurance that organizations are able to specify their own policy.

In addition to what is mentioned above, SDN characterizes with other advantages such as:

- Network's centralization: The SDN adopts the centralized monitoring and management of the network, as well as to the centralized security.
- Hardware optimization: The SDN reduce the use of physical hardware by the orientation toward virtualized network infrastructures. When we say optimization of the hardware, we say coast reduction as well.

2 SDN Architecture

The SDN architecture is characterized by the separation of control plane from the data plane. Control plane is the brain who takes the decision of traffic networking. Data plane or forwarding plane is responsible of forwarding traffic, according to the control plane to the next component. The architecture of SDN contains three layers namely application layer, controller layer and infrastructure layer as it's showed in Fig. 1.

2.1 Controller Layer

Controller layer contains a bunch of controllers that are responsible of controlling the network. In other words, the controller layer is the control plane which is the principal component that takes the decision about the optimal path that traffic will take and monitor the behavior of the forwarding network. The controller uses protocols in order to configure the network devices such as OpenFlow [4, 10, 15].

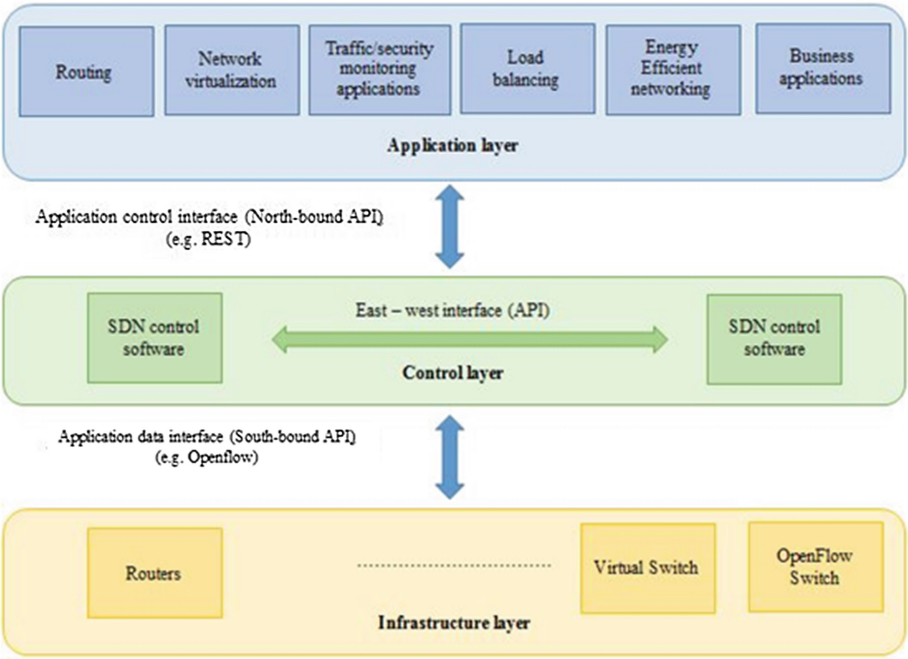

Fig. 1. SDN's architecture

Controllers communicate between them through east - west interfaces in order to maintain the synchronization and connectivity of the network [2, 3, 6, 8]. The controller layer communicates with other layers using north-bound API's and south-bound API's. Where, north-bound API's (e.g. REST, frantic, etc.) are used to communicate with application layer and south-bound API's (e.g. OpenFlow, NetConf, etc.) to communicate with infrastructure API's [8–10].

2.2 Application Layer

The application layer is built on the controller layer, which represents the first layer in SDN architecture. It contains a set of software related to business requirements such as intrusion detection systems (IDS) [11], network virtualization [12], load balancing [13], and so on. In the case of changes at the application layer, controller layers afford an abstraction of network's resources to be allocated to the software of the application layer, in order to avoid reconfiguration of the network's resources such as switches and routers.

2.3 Infrastructure Layer

The infrastructures layer is also known as the data forwarding layer, it contains virtual or physical network resources and devices. As its name mentions, it's responsible for the forwarding of packets of the network according to a set of rules within the flow

table [10, 14, 15]. The flow table entries contain three section namely the pattern, action, and stats [2]. The pattern represents the header field of a packet; the action is executed according to the match of the rules, then stats, which are indications that indicates the network's status.

3 Security Issues

In this section, we present a set of security issues of different layers that may lead to a successful attack.

Open programmable APIs: As it's mentioned above, SDN communicates through programmable APIs, these APIs can be open which may cause security issues by making the layers open and the vulnerabilities of components of the SDN visible to attackers. This issue may lead to cross-site scripting attack (XSS) or injection of malicious code [16, 17].

The controller issues: Because of the central architecture of SDN, the configuration and the decision of the network is taken by the controllers. Therefore, an exploitation of vulnerability can gain the attacker to take control of the whole network which can cause huge damages [11].

The SDN switches issues: switches within SDN suffer from the limitation of entries of the flow table. This issue makes switches very sensitive to DDoS attacks.

4 SDN Attacks

SDN attracts attackers to look for vulnerabilities in order to use them to exercise attacks or a set of attacks. In this section, we classify the attacks according to target the layer.

Figure 2 shows the different attack point in the SDN architecture, which an attacker can exploit the existed vulnerability [11].

An attack can be exercised on the component of the application layer, against controllers of controller layer and at channels of communication between controllers. Switches are not excluded from these attacks. It also, can target the programmable API's that connects layers to each other.

4.1 Application Layer

The application layer may contain vulnerabilities related to software and the difficulties of modeling a global security policy that is able to manage the whole network without fails. Where most of the applications are developed by a third party, which doesn't take into consideration the mechanism of security standardization.

The rest of this section describes the major attacks faced by the application layer.

Unauthorized access: The large number of devices of a network may lead to the misuse of the application running on controllers by an intruder to gain unauthorized access to sensitive information such as network information. As most of the applications are made

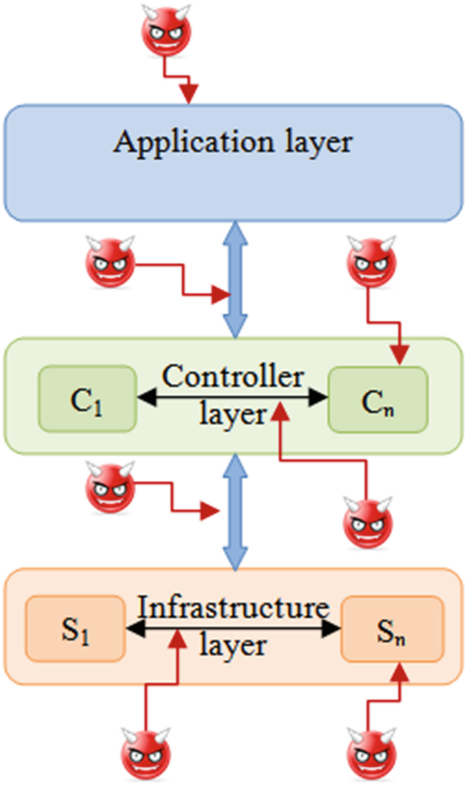

Fig. 2. Attack point in SDN architecture

by third parties that have a limited knowledge about the security requirement such as the management of authentication systems, authorized accesses to an application, and the access of applications to the network information [11].

Malicious programs and application injection: Code, programs or applications injection is one of top ten attacks that target applications especially web applications according to OWASP project [16]. This attack may cause unauthorized access, data loss or information corruption. It's used by worms to propagate within the network. Also, it helps attackers to gain more privileges to accomplish their malicious tasks.

Insertion of rules: In order to get the benefit of the advantages of SDN, SDN paradigm is applied in various areas namely cloud computing, data centers, cellular networks, wireless networks, mobile networks, etc. where the number of devices is huge, with complex applications and services. Therefore, The insertion and the management of security rules is a big challenge for administrators and providers in order to prevent security rules conflicts between applications and services [5, 6, 11].

4.2 Controller Layer

The controllers are the brain of the SDN. Thus, because of its importance, attackers aim to get control of the whole network by exploiting the existing vulnerabilities. This section presents the well-known attacks faced by controllers.

Attacks from application layer: Applications are running on controllers, were any successful attack on application layer may lead to security issues in the controller layer. For example, application injection attack can gain access to network devices information and monitor the behavior of the network, or exercise other attacks for more serious effects.

DDoS/ DoS attack: Denial of service (DoS) and distributed DoS (DDoS) is the simplest attack exercised by attackers that target the availability of the network and services for the legitimate users [11]. This attack consumes the controller's resource such as CPU, memory, and bandwidth by rules installation and computation from the flooded flow requests [18]. Once the controller is saturated, the legitimate requests will be dropped and the switches connected to the affected controller will be affected as well [18].

Attacks against distributed multi-controllers: Because of the division of the main network into sub-networks, the need of using distributed multi-controller raises. This solution was proposed to overcome the DDoS/ DoS attacks and preventing the shutting down of the whole network. However, the SDN remains sensitive to DDoS/ DoS attacks, and to other issues related the management of the security policy and security conflict [11, 18].

4.3 Infrastructure Layer

Switches within infrastructure layer are divided into three part especially the OpenFlow switches: OpenFlow agent, packet buffer and table flow, which are a target for DoS attack.

DoS Attack: To perform the DoS attack, the attacker performs "the flow request flooding" by interrupting the performance of the three parts of OpenFlow switch. He/she sends a large number of malformed packets to saturate the OpenFlow agent since it generates a limited number of flow requests per second to be sent to the controller. Thus, the target switch is affected as well as to the hosts connected to the victim. In the case of a full packet buffer, the victim switch sends instead of packets headers, the entire packets to the controller that lead to the consumption of the bandwidth and channel congestion [11, 18]. Another drawback of the OpenFlow switch is the limited entries of a flow table, where the attacker aims to overflow it by installing new rules. This attack leads to dropping rules of legitimate flow [20].

Man-in-the-middle: The attacker of man-in-the-middle (MITM) monitors the traffic between controllers and switches, in order to intercept the information of communication without being detected. Controllers and switches are not directly connected to each other, which makes each entity doubtful to be a MITM node [19]. MITM attack

Table 1. The proposed solution to sdn security issues [11, 42]

SDN layer	Security issue	Proposed solution	Description
Application layer	Unauthorized access	OperationCheckpoint [23]	Presents a permission checkpoint to verify the authorization of applications
		SE-Floodlight [24]	Includes the digital authenticated northbound API for a minimum privilege
		AuthFlow [25]	Presents authentication and access control mechanism based on host credential
		NICE [22]	Verifies the correctness of OpenFlow application by automated the testing
		VeriCon, Verificare [26]	Verify the correctness of controller's applications and verifies the correctness of execution of any single network event
	Malicious programs and application injection	FortNox [27]	Provides role-based authorization and security constraint enforcement for the NOX OpenFlow controller
		LegoSDN [28]	Introduces fault-tolerant controller framework that allows SDN controllers to isolate and tolerate SDN application failures, in order to increase the availability of the network
		ROSEMARY [29]	Implements a network application containment and resilience strategy based on the notion of spawning applications independently
	Insertion of rules	NetPlumber [30]	Presents real-time policy checking and incrementally checks for compliance of state changes, using a dependency graph between rules
		Anteater [31]	Diagnosis problems through static analysis of the data plane in order to identify policy conflicts

(*continued*)

Table 1. (*continued*)

SDN layer	Security issue	Proposed solution	Description
		Flover [32]	Introduces a model of checking system which verifies that the aggregate of flow policies instantiated within an OpenFlow network does not violate the network's security policy
Controller layer	Attacks from application layer	SE-Floodlight [24]	Tracks the event flow of application to detect any attack that may come from applications
		FRESCO [33]	Implements **different** security function such as firew**alls, scan detectors, attack de**flectors, or IDS detection logic
	DDoS/DoS attack	FloodGuard [34]	**Introduces a scalable, efficient** and lightweight framework for SDN networks to prevent data-to-control plane saturation attack by using packet migration and data plane cache.
		CONA [35]	Analysis the content of requests made by the client to a server in order to reduce the harm of DDoS/DoS attacks
	Distributed multi-controllers	HyperFlow [36]	HyperFlow localizes decision making to individual controllers, thus minimizing the control plane response time to data plane requests
		McNettle [37]	Presents an extensible SDN control system based on multi-cores CPUs to control event **processing. The** processing of events related to the number of CPU cores
		DISCO [38]	Presents a distributed DISCO controller, where each one manages its own domain and communicates to each other to share and provide network services

(*continued*)

Table 1. (*continued*)

SDN layer	Security issue	Proposed solution	Description
Infrastructure layer	DDoS/DoS attack	VAVE [39]	Provides a solution that verifies the validity of source address that causes DoS attack
		FlowVisor [40]	Presents a switch virtualization, where the same hardware forwarding plane can be shared between various logical networks, each with a distinct forwarding logic
	Man-in-the-middle	VeriFlow [41]	Presents a layer between a software-defined networking controller and network devices, and supports analysis over multiple header fields, and an API for checking custom invariants
		FortNox [27]	Verifies the legitimacy of the modifications through digital signatures or security constraints

leads to the implementation of other attacks such us eavesdropping and black-hole attack [11, 20].

5 Countermeasures of SDN Attacks

This section deals with the solution that has been proposed to solve some of SDN security issues mentioned above. The following table (Table 1) summarizes the proposed solution with a description.

6 Conclusion

In this paper, we presented a review of the security in the software-defined system. The first part describes the different components of SDN architecture with their characteristics. The second part contains the security issues and a list of attacks faced by the elements of SDN. And in the third part, we gave a set of proposed solutions that aim to solve or mitigate the harm of attacks, these solutions are divided according to the three layer: Application layer, the controller layer, and infrastructure layer. DDoS/ DoS attack is one of the most common attacks that target the SDN at different levels (Application layer, the controller layer, and infrastructure layer).

In our future research, we intend to concentrate on the lack of visibility of the SDN state within cloud computing by proposing an approach that measures the security state of the virtual network and provides the appropriate countermeasure in case of an attack.

Referernces

1. Khalil, I.M., Khreishah, A., Azeem, M.: Cloud computing security: A survey. Computers **3**(1), 1–35 (2014)
2. Gong, Y., Huang, W., Wang, W., Lei, Y.: A survey on software defined networking and its applications. Front. Comput. Sci. **9**(6), 827–845 (2015)
3. Cisco Inc.: Software-defined networking: why we like it and how we are building on it, White Paper (2013)
4. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Turner, J.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
5. Ahmad, I., Nama, S., Ylianttila, M., Gurtov, A.: Security in software defined networks: A survey. IEEE Commun. Surv. Tutorials **17**(4), 2317–2346 (2015)
6. Rawat, D.B., Reddy, S.R.: Software defined networking architecture, security and energy efficiency: A survey. IEEE Commun. Surv. Tutorials **19**(1), 1–22 (2016)
7. Kemmer, F., Reich, C., Knahl, M., Nathan, C.: Software defined privacy. In: IEEE International Conference on Cloud Engineering Workshop, pp. 25–29 (2016)
8. Han, B., Gopalakrishnan, V., Ji, L.S., Lee, S.J.: Network function virtualization: challenges and opportunities for innovations. IEEE Commun. Mag. **53**(2), 90–97 (2015)
9. Yang, W., Fung, C.: A survey on security in network functions virtualization. In: IEEE NetSoft Conference and Workshops (NetSoft), pp. 15–19 (2016)
10. Hu, F., Hao, Q., Bao, K.: A survey on software-defined network and OpenFlow from concept to implementation. IEEE Commun. Surv. Tutorials **16**(4), 2181–2206 (2014)
11. Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A.V., Imran, M.: Security in software-defined networking: Threats and countermeasures. Mobile Netw. Appl. **21**(5), 764–776 (2016)
12. Bernardo, D.V.: Software-defined networking and network function virtualization security architecture (2017). https://tools.ietf.org/html/draft-bernardo-sec-arch-sdnnvf-architecture-00
13. Namal, S., Ahmad, I., Gurtov, A., Ylianttila, M.: SDN based intertechnology load balancing leveraged by flow admission control. In: IEEE SDN for Future Networks and Services, pp. 1–5 (2013)
14. Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: A Comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
15. Stallings, W.: Software-defined networks and OpenFlow. Internet Protoc. J. 16 (2015)
16. Top ten web application vulnerabilities (2017). https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
17. Green, M., Smith, M.: Developers are not the enemy!: the need for usable security APIs. IEEE Secur. Priv. **14**(5), 40–46 (2016)
18. Zhang, P., Wang, H., Hu, C., Lin, C.: On denial of service attacks in software defined networks. IEEE Netw. **30**(6), 28–33 (2016)
19. Brezetz, S.B., Kamga, G.B., Balla, M.N., Criton, T., Jebalia, H.: SDN-based trusted path in a multi-domain network. In: IEEE International Conference on Cloud Engineering Workshop, pp. 19–24 (2016)

20. Benton, K., Camp, L.J., Small, C.: OpenFlow vulnerability assessment. In: 2nd ACM SIGCOMM workshop on Hot Topics in Software Defined Networking, pp. 151–152 (2013)
21. Wen, X., Chen, Y., Hu, C., Shi, C., Wang, Y.: Towards a secure controller platform for openflow applications. In: The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 171–172 (2013)
22. Canini, M., Venzano, D., Peresini, P., Kostic, D., Rexford, J.: A NICE way to test OpenFlow applications. In: The 9th USENIX Conference on Networked Systems Design and Implementation (2012)
23. Yu, D., Moore, A.W., Hall, C., Anderson, R.: Authentication for resilience: The case of SDN. In: Security Protocols XXI. Springer, Berlin, pp. 39–44 (2013)
24. Security Enhanced (SE) Floodlight (2017). http://www.openflowsec.org/Technologies.html
25. Mattos, D.M.F., Ferraz, L.H.G., Duarte, O.C.M.B.: AuthFlow: Authentication and access control mechanism for software defined networking. Univ. Federal Rio Janeiro, Rio de Janeiro, Brazil (2014)
26. Ball, T., Bjmer, N., Gember, A., Itzhaky, S., Karbyshev, A., Sagiv, M., Valadarsky, A.: Vericon: towards verifying controller programs in software-defined networks. ACM SIGPLAN Not. **49**(6), 282–293 (2014)
27. Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., Gu, G.: A security enforcement kernel for OpenFlow networks. In: 1st Workshop Hot Topics Software Defined Network, pp. 121–126 (2012)
28. Chandrasekaran, B., Benson, T.: Tolerating SDN application failures with LegoSDN. In: Proceedings of the 13th ACM Workshop Hot Topics Network (2014)
29. Shin, S., et al.: Rosemary: A robust, secure, and high-performance network operating system. In: ACM Conference on Computer and Communications Security, pp. 78–89 (2014)
30. Kazemian, P., Chan, M., Zeng H., Varghese, G., McKeown, N., Whyte, S.: Real time network policy checking using header space analysis. In: USENIX Symposium on Networked Systems Design and Implementation, pp. 99–111 (2013)
31. Mai, H., Khurshid, A., Agarwal, R., Caesar, M., Godfrey, P., King, S.: Debugging the data plane with anteater. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 290–301 (2011)
32. Son, S., Shin, S., Yegneswaran, V., Porras, P., Gu, G.: Model checking invariant security properties in OpenFlow. In: International Conference on Communications (ICC), pp. 1974–1979 (2013)
33. Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M.: FRESCO: Modular composable security services for software-defined Networks. In: Network and Distributed Security Symposium, pp. 1–16 (2013)
34. Wang, H., Xu, L., Gu, G.: FloodGuard: a dos attack prevention extension in software-defined networks. In: 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 239–250 (2015)
35. Suh, J., Choi, H. G., Yoon, W., You, T., Kwon, T., Choi, Y.: Implementation of a content-oriented networking architecture (CONA): a focus on DDoS countermeasure. In: European NetFPGA Developers Workshop (2010)
36. Tootoonchian, A., Ganjali, Y.: HyperFlow: a distributed control plane for OpenFlow. In: The 2010 Internet Network Management Conference on Research on Enterprise Networking. USENIX Association, p. 3 (2010)
37. Voellmy, A., Wang, J.: Scalable software defined network controllers. In: The ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 289–290 (2012)

38. Phemius, K., Bouet, M., Leguay, J.: DISCO: Distributed SDN controllers in a multi-domain environment. In: IEEE Network Operations and Management Symposium (NOMS), pp. 1–4 (2014)
39. Yao, G., Bi, J., Xiao, P.: Source address validation solution with OpenFlow/NOX architecture. In: 19th IEEE International Conference on Network Protocols (ICNP), pp. 7–12 (2011)
40. Sherwood, R., Gibb, G., Yap, K.K., Appenzeller, G., Casado, M., McKeown, N., Parulkar, G.: Flowvisor: a network virtualization layer. OpenFlow Switch Consortium, Technical Report (2009)
41. Khurshid, A., Zhou, W., Caesar, M., Godfrey, P.: Veriflow: verifying network-wide invariants in real time. In: ACM SIGCOMM Computer Communication Review, pp. 467–472 (2012)
42. Scott-Hayward, S., Natarajan, S., Sezer, S.: A survey of security in software defined networks. IEEE Commun. Surv. Tutorials **18**(1), 623–654 (2015)