NUST Journal of Engineering Sciences, Vol. 8, No. 1, 2015, pp. 38-43

# Benefits, Security and Issues in Software Defined Networking (SDN)

**Nasir Shahzad*, Ghulam Mujtaba, Manzoor Elahi**

Department of Electrical Engineering, Comsats Institute of Information Technology, Pakistan
*Corresponding author: mnasirshahzad@ciit.net.pk

## Abstract

Applications used now a day are bandwidth hungry like Online Shopping, IPTV, E-Commerce and many other which require more and more bandwidth as well as continuous bandwidth. The Software Defined Networking (SDN) decouples control and forward plane allowing the flexibility to program network control plane and empowers distinctive approaches to network security than those existing in present IP system. In SDN, the centralized controllers keep an eye on the changing scenarios of the network. Because of controller view of the network, SDN can facilitate and enhance the network related security. SDN architecture is directly programmable and opens standard-based but SDN itself have numerous issues like performance vs. flexibility, scalability, security and interoperability. This paper discusses security issues regarding logically centralized controller, OpenFlow constraints and absence of middle-boxes in SDN.

**Keywords:** Software Defined Networking, Architecture, Benefit of SDN, Security.

## Introduction to SDN Architecture

SDN is promising network design which is specifically programmable with network control decoupled from forwarding. This transfer of control, once firmly guaranteed in individual network devices, into open figuring devices empowers basic foundation to be absorbed for applications and network administrations that treat the system as a legitimate element.

Figure 1 delineates a coherent perspective of the SDN structural engineering. Network sagacity is (intelligently) brought together in programming, which keeps up a worldwide perspective of the system. With SDN, ventures and bearers pick up merchant autonomous control over the whole system from a solitary consistent point which incredibly streamlines the system outline and operation. SDN likewise incredibly improves the network devices themselves, since they probably won't have to comprehend and process a large number of protocol models yet simply acknowledge directions from the SDN controllers. SDN gives network manager the adaptability to design, oversee, secure and enhance system assets by means of dynamic, computerized SDN programs.

In like manner, SDN marks it conceivable to deal with the whole network through adroit coordination and provisioning frameworks. The Open Networking Foundation is considering open APIs to advance multi-vendor administration, opening entryway for on-interest asset allotment, organization toward oneself provisioning, positively virtualized network administration and secure cloud services. SDN makes the system less "application-aware" but rather more "application-customized" and applications less "network-aware" but rather more "network-capability aware".

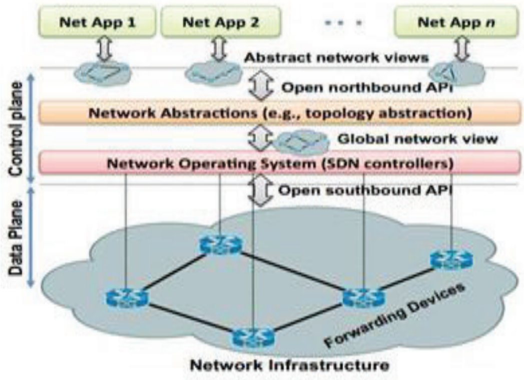

**Fig-1. Secure and dependable SDN**

## Terminologies

We present some terminologies to comprehend different elements of SDN network.

### Forwarding Devices (FD)

It is programming centred data plane devices performing basic operations through decently characterized instruction sets (flow rule). These instructions are characterized by southbound interfaces.

### Southbound Interface (SI)

API characterizes the guideline set of sending devices which is a piece of the southbound interface. The SI likewise characterizes the protocol between sending devices and control plane.

### Management Plane (MP)

The management plane influence the capacities offered by the NI to actualize control and operation rationale. This incorporates applications, for example, steering, firewalls, load balancers, **checking, etc.**

## Benefits of Open Flow-Based Software Defined Networks

The benefits achieved through OpenFlow-based Software Defined Networking are:

### Centralized control of multi-vendor environments: SDN controls any OpenFlow-empowered devices like switches and virtual switches from any vendor

Instead of needing to oversee gatherings of devices from individual vendors, IT can utilize SDN-based coordination and administration devices to rapidly send and arrange devices over the whole system.

### Reduced Complexity by Automation

OpenFlow-based SDN offers an adaptable network automation and administration structure, which makes it conceivable to create apparatuses automating numerous administration undertakings that are carried out physically. These mechanization devices decrease operative overhead, decline system insecurity presented by administrator mistake, and help developing IT as a service and organization toward oneself provisioning models.

### The Higher rate of innovation

SDN selection quickens business advancement by permitting IT to arrange administrators to truly program and reconstruct system progressively to meet particular business needs and client prerequisites as they emerge. For instance, SDN and OpenFlow feed IT and conceivably even clients the capacity to modify the conduct of the system and present new administrations and system abilities in hours.

### Increased network reliability and security

SDN makes it possible for IT to characterize abnormal state arrangement and strategy proclamations, which are then made an interpretation of down to the framework through OpenFlow. An OpenFlow-based SDN structural planning takes out the necessity to independently design devices, administration, or application is included, or an approach alteration, which decreases the probability of network failure because of arrangement or strategy inconsistencies. SDN controllers can guarantee that get to control, traffic engineering, nature of administration, security, and different strategies are implemented reliably over the wired and remote system frameworks, including extension business locales, grounds, and server farms. Enterprises and carriers get advantage from diminished operational costs, more dynamic arrangement capacities, fever less slips, and reliable design and approach authorization.

### More granular network control

Open Flow's stream permits IT to apply strategies at an exceptionally granular level, involving client, devices and application levels, in a very preoccupied, mechanized design. This control empowers cloud administrators to help militancy while keeping up activity detachment, security and flexible asset administration when clients have the same framework.

### Better client experience

A bearer could present a feature benefit that offers premium supporters the most noteworthy conceivable determination in a robotized and straightforward way. Today, clients should expressly select a determination setting, which the system might possibly have the capacity to help, bringing about postponements and interferences that corrupt the client experience. With OpenFlow-based SDN, the feature application would have the capacity to identify the data transmission accessible in the network progressively and consequently modify the feature determination likewise.

## Some security issues in SDN and proposed solutions

### Logically centralized controller

Software-defined network administration enables system administrators with more adaptability to program their network. For example, while at the same time permitting the utilization of security and constancy methods. The logically centralized controlled plane causes a stage for attacker which brings about a few primary potential risk vectors we distinguished in SDNs. Our objective is not to utilize these potential issues to the case that product characterized systems are characteristically less safe than current systems.

### Fake flow

This technique is utilized to access switches and controllers? This danger can be activated by flawed devices or clients. An intruder can utilize network components to dispatch a DoS assault against OpenFlow switches and controller assets. A basic verification instrument could relieve the issue, yet in the event that an assailant accepts the control of a server that stores the points of interest of numerous clients, it can undoubtedly utilize the same confirmed ports and source MAC locations to infuse approved, yet fashioned, streams into the system.

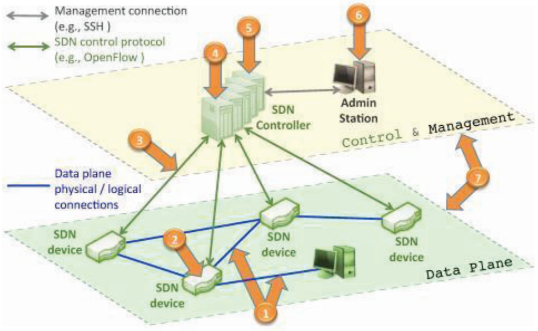

**Fig-2. SDN main threat vector map**

**Possible solution:** The use of intrusion detection systems [2] with support for runtime root-cause analysis could help identify abnormal flows. This could be coupled with instruments for element control of switch conduct. This issue can be stayed away from by intrusion detection system to distinguish the strange stream.

### The Attack on control plane communication

Because of shortcomings TSL/SSL-causes fake stream of information theft. TLS/SSL does not as such ensure secure communication [3, 4] and that bargains the controller-device join. The security of that communication is as solid as its weakest connection, which could be a checked toward oneself corticated, a traded off Corticated Authority, or vulnerable applications and libraries. For example, here is numerous man-in-the-center defenceless usage of SSL being utilized as a part of overall discriminating frameworks [5]. Also, the TLS/SSL model is insufficient to build and guarantee trust in the middle of controllers and switches. Once an attacker gets access to the control plane, it might be fit for collecting enough power force to dispatch DDoS attacks [6, 7]. This absence of trust certifications could even empower the formation of a virtual black hole network [8] permitting information spillage while the ordinary creation traffic streams.

**Possible solution:** The utilization of oligarchic trust models with different trust anchor certification authorities is a possibility. An alternate is securing correspondence with limit cryptography crosswise over controller replicas [9].

### The Attack on the vulnerabilities in the controller

Which are presumably the most serious dangers to SDNs? A defective or malicious controller could trade off a whole network. The utilization of a typical intrusion detection system may not be sufficient; as it might be elusive the precise blend of occasions that trigger a specific conduct and all the more imperatively to name it as pernicious. Essentially, a pernicious application can conceivably do anything it satisfies in the network since controllers just give reflections that interpret into issuing design summons to the basic foundation.

**Possible solution:** A few procedures can be utilized to stay away from this issue, for example, replication (to distinguish, evacuate or cover irregular conduct), utilizing differences and recuperation (intermittently reviving the framework to a clean and dependable state). It is likewise paramount to secure all the sensitive components inside the controller (e.g., crypto keys/secrets).

### Lack of a mechanism to ensure trust between controller and management application

It causes the malevolent application to be sent to the controller.

**Possible solution:** To keep away from this, instruments for autonomic trust administration could be utilized to ensure that the application is trusted amid its lifetime.

### Lack of trusted resources for forensics and remediation

Which would permit to comprehend the reason for a distinguished issue and move ahead too quick and secure mode recuperation. Keeping in mind the end goal to examine and make realities about an episode, we require dependable data from all parts and areas of the network. Moreover, this information might be valuable on the off chance that its dependability (trustworthiness, validness, and so on) can be guaranteed. Correspondingly, remediation obliges protected and dependable framework depictions to ensure a quick and right recuperation of system components to a known working state.

**Possible solution:** Logging and tracing are the common mechanisms in use and are required both in the information and control planes. Then again, with a specific end goal to be viable, they ought to be permanent. Moreover, logs ought to be put away in the remote and secure environment.

### Centralized networking operation and security

From many years traditional networking administration accumulation information from all networking nodes is done by different protocols, e.g. SNMP [10] and NetFlow [11]. However, to get to information in traffic is not a simple errand in light of the fact that to get data, numerous queries are required which obliges expensive administration and CPU usage. At the same time in SDN, it is a simple errand on account of incorporated nature of SDN, a worldwide perspective of systems administration helps us to locate the appropriated DOS attacks which are inconceivable in a solitary node [12]. The second point of interest of worldwide perspective is connected with SDN controller capacity to control each one stream sent however NW any response to located risk is prompt. However, this centralized architecture has a few issues moreover. Since,

1. A vast amount of network flow must be analyzed in one place.
2. Introduce a single point of failure which leads to congestion to SDN controller.
3. Malicious users can deliberately generate traffic to disturb an SDN NW [13].

Any successful attack on the centralized controller can cause severe network degradation.

**Possible solution:** Logical distribution of physical controllers may ease this risk to some degree, yet a fastidious security of control assets is basic. The insurance ought to cover all angles – not always technical but "social". In a legacy network, this sort of risk is not generally discriminating and effect of a solitary security rupture can be contained. A deliberately thoroughly considered system outline (e.g. directing arrangements, OSPF ranges, individual connection security) is the answer for security authorization of today IP networks.

### Finger-printing the SDN [14]

In SDN the control of information planes is separated from one another because of which there are numerous security issues confronting by SDN, despite the fact that the SDN architecture has numerous advantages as contrasted with the previous version of NW, at the same time there are numerous blemishes connecting with it. Because of the sensitive mode of the control plane, the reaction to numerous solicitation will result in the flooding of flow table and additionally help the attacker to unique mark the NW for fingerprinting t-test [15, 16] technique is utilized. For finger-printing, the attacker uses SDN scanner. In the event that an attacker runs SDN scanner and gathers network data, he/she can explore whether a target network is utilizing SDN or not through a straightforward measurable testing technique.

### OpenFlow limitations in the context of security [17]:

OF has picked up a predominant position that it advances more protocol headers yet it has a few downsides, the limits connected with the strict meaning of field utilized by sending components. For example, Ipv6 fields have not presented in OFv1.3. In this way to change the field of distinctive conventions in OF fit switches empowers more entangled capacities than sending i.e. NAT or firewall.

**Possible solution:** Protocol Oblivious Forwarding (POF) [18] gives more adaptability and focal points. Conversely, the OpenFlow, the POF adaptability permits for fast advancement and usage of new protocols, with no changes in the switch fittings and no alterations in correspondence with the controller. Besides, applications that actualize new SDN administrations utilizing POF can settle on choices utilizing any ingress packet. This gives a colossal adaptability and unbelievable these days usefulness of future SDN empowered systems. Particularly for security related applications, this adaptability can be useful, permitting usage of a "Deep packet inspections" [19].

### Lack of middle-boxes in SDN [20]

OF avoid execution of some function e.g. deep packet inspection. It is not optimal from an execution perspective, to process all choice concerning each stream by the SDN controller. A decentralization of some SDN functions, despite the fact that it breaks the SDN standard, can lead to more productive and versatile network. The decentralized capacities can be performed generally on SDN switches. Usage of security capacities in this spot has numerous points of interest. This can enhance discovery rate in correlation to activity watched in collected joins claimed by ISP. On the other hand, lack of middle boxes in the architecture definition can imply deficiencies in security.

**Possible Solution**: Simple Intrusion Detection Systems (IDS) analyze signatures or anomalies, more advanced ones utilize data exploration algorithms. Distributed Frequent Sets Analyzer (DFSA) [21] systems are a good approach. The DFSA system takes advantage from experiments with anomaly detection using data mining [22, 23] and from the features of SDN network.

Our first need is to secure the controller (the most adroit part). So there are two strategies to keep away from the security challenges. One is NetFuse and Fresco.

## FRESCO [24]

It is intended to encourage the fast plan, and particular piece OF-empowered identification and moderation modules. FRESCO, which is itself an OpenFlow application, offers a Click-inspired [25] programming framework. Fresco gives a stage a bundle of valuable data about security. Their intentions are to (i) Script to module interpretation (ii) Database administration (iii) Event administration (iv) Instance execution (v) The security authorization piece screen and keep the tracks of switches in standard interim. FRESCO is proposed to address a few key issues that can quicken the synthesis of new OF-empowered security administrations. FRESCO trades a scripting API that empowers security specialists to code security checking and risk discovery rationale as measured libraries. These secluded libraries speak to the primary handling units in FRESCO, and may be imparted and connected together to give complex network resistance applications.

FRESCO as of now incorporates a library of 16 normally reusable modules, which we expect to grow after some time. Conceivably, more advanced security modules can be manufactured by joining essential FRESCO modules. Each one FRESCO module incorporates five interfaces: (i) include, (ii) yield, (iii) occasion, (iv) parameter and (v) activity. By essentially relegating qualities to every interface and joining fundamental modules, a FRESCO designer can repeat a scope of vital security capacities, for example, firewalls; examine identifiers, assault diverters, or IDS identification rationale.

## Conclusion

Because of the changing nature of internet traffic, network engineers have to accommodate the bursty traffic. Bandwidth consuming applications like IPTV, Online games and online banking require continuous services. SDN being dynamic architecture provides cost-effectiveness, adaptability and management. Separation of Control and Forwarding plane enables the control plane to be directly programmed. The changing needs of the network are adjusted by the administrator on the go. The administrator having the ability to program the network according to the conditions raises a question on security itself. So security of SDN architecture is an open question for researchers. In this paper, we have assessed a portion of the security related issues, for example, Denial of Service, information disclosure etc., which are aggravated due to the nature of SDN that degrades the execution.

Basic idea is to analyze the network statistics from the forwarding plane using standardized methods and apply classification algorithms to detect any irregularity like using OpenFlow. In view of our analysis and assessment, we proposed distinctive arrangement relating to diverse issues which can be adjusted to future variant and expansions of OpenFlow and to develop our own software-defined networking to address the above security challenges.

## Acknowledgements

The authors are grateful to Dr. Hazrat Ali, CIIT Abbottabad, for his useful comments on this manuscript.

## References

1. N. Gude, et al., "NOX: towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, 2008, pp. 105-110.

2. M. Scheidell, "Intrusion detection system," *Book Intrusion detection system*, Series Intrusion detection system, ed., Editor ed.^eds., Google Patents, 2009, pp.

3. S. Raza, et al., "Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN," *Security and Communication Networks*, vol. 7, no. 12, 2014, pp. 2654-2668.

4. H. Suo, et al., "Security in the internet of things: a review," *Proc. Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, IEEE, 2012, pp. 648-651.

5. M. Georgiev, et al., "The most dangerous code in the world: validating SSL certificates in non-browser software," *Proc. Proceedings of the 2012 ACM conference on Computer and communications security*, ACM, 2012, pp. 38-49.

6. S.A. Mehdi, et al., "Revisiting traffic anomaly detection using software defined networking," *Proc. International Workshop on Recent Advances in Intrusion Detection*, Springer, 2011, pp. 161-180.

7. R. Braga, et al., "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *Proc. Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, IEEE, 2010, pp. 408-415.

8. R. Sherwood, et al., "Flowvisor: A network virtualization layer," *OpenFlow Switch Consortium, Tech. Rep*, 2009, pp. 1-13.

9. S. Goldwasser, et al., "Cryptography and Information Security Group Research Project: Threshold Cryptology," *Book Cryptography and Information Security Group Research*

*Project: Threshold Cryptology*, Series Cryptography and Information Security Group Research Project: Threshold Cryptology, ed., Editor ed.^eds., 2013, pp.

10. V. Cerf, et al., *Delay-tolerant networking architecture*, 2070-1721, 2007.
11. N. McKeown, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, 2008, pp. 69-74.
12. C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network security*, vol. 2011, no. 8, 2011, pp. 16-19.
13. D. Kreutz, et al., "Towards secure and dependable software-defined networks," *Proc. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, ACM, 2013, pp. 55-60.
14. S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," *Proc. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, ACM, 2013, pp. 165-166.
15. Y. Wang, et al., "NetFuse: Short-circuiting traffic surges in the cloud," *Proc. 2013 IEEE International Conference on Communications (ICC)*, IEEE, 2013, pp. 3514-3518.
16. K. Govindarajan, et al., "A literature review on software-defined networking (SDN) research topics, challenges and solutions," *Proc. 2013 Fifth International Conference on Advanced Computing (ICoAC)*, IEEE, 2013, pp. 293-299.
17. G. Stabler, et al., "Elastic IP and security groups implementation using OpenFlow," *Proc. Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date*, ACM, 2012, pp. 53-60.
18. C. Boldrini, et al., "Modelling social-aware forwarding in opportunistic networks," *Performance Evaluation of Computer and Communication Systems. Milestones and Future Challenges*, Springer, 2011, pp. 141-152.
19. M. Becchi, et al., "A workload for evaluating deep packet inspection architectures," *Proc. Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*, IEEE, 2008, pp. 79-89.
20. S.K. Fayazbakhsh, et al., "Enforcing network-wide policies in the presence of dynamic middlebox actions using flowtags," *Proc. 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 543-546.
21. D. Remane, et al., "Development and validation of a liquid chromatography-tandem mass spectrometry (LC-MS/MS) procedure for screening of urine specimens for 100 analytes relevant in drug-facilitated crime (DFC)," *Analytical and bioanalytical chemistry*, vol. 406, no. 18, 2014, pp. 4411-4424.
22. Z. Zhao, et al., "SAHAD: Subgraph analysis in massive networks using Hadoop," *Proc. Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, IEEE, 2012, pp. 390-401.
23. K. Savitha and M. Vijaya, "Mining of web server logs in a distributed cluster using big data technologies," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 5, no. 1, 2014.
24. S. Shin, et al., "FRESCO: Modular Composable Security Services for Software-Defined Networks," *Proc. NDSS*, 2013.
25. H. Farhadi, et al., "Enhancing OpenFlow actions to offload packet-in processing," *Proc. Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, IEEE, 2014, pp. 1-6.