THE USING OF SDN TECHNOLOGIES FOR SECURITY INSURANCE OF COMPUTER NETWORKS

ALI AMEEN

Technical University of Moldova

Abstract: *software-defined networking is a new technology that affects different aspects and fields of information technology and is used to provide a better security for networks environment and uses different other technologies to achieve this goal. Due to the growing data and the need to accommodate with these data, also as we mentioned earlier to secure networks because of the ever-evolving security threats we have to find a new technology or method to match the growing threats. This article provides a simple analysis for the structure of SDN and gives some information about the history of SDN and related work that shows the effect of SDN in the world of technology. Then we'll talk about the pros and cons of software-defined networks, after that comes the simple description for some ideas and proposed topologies and algorithms. And we attach a simple conclusion about this article.*

Keywords: *control plane, data plane, DDoS, NFV, ONF, programming*

Introduction

Networking field hasn't seen much development since the 80's after the invention of OSI model, and since this field is the essence of mostly used technology in today's life which is the internet, it is of a great deal of necessity to develop it to be able to accommodate with the other ever evolving technologies. Also, due to the huge growth of data centers and clouds we got another issue which is dealing with these huge amounts of data resulting from these technologies alongside with other technologies like the emergence of IoT technology and the usage of smartphones…etc.

All that led to the need to develop networks to be able to contain and work with these data and information; so, a few years ago came the invention of software-defined networks or (SDN), which simply means adding the ability of programmability to networks instead of the old rigid structure of usage where we have some preconfigured vendor-dependent devices that constrain us in the form of our developed needs. In the classical network, devices can be configured to a simple limited extent by the network administrator but, with SDN admins can accommodate with the ever-evolving needs of enterprises and datacenters.

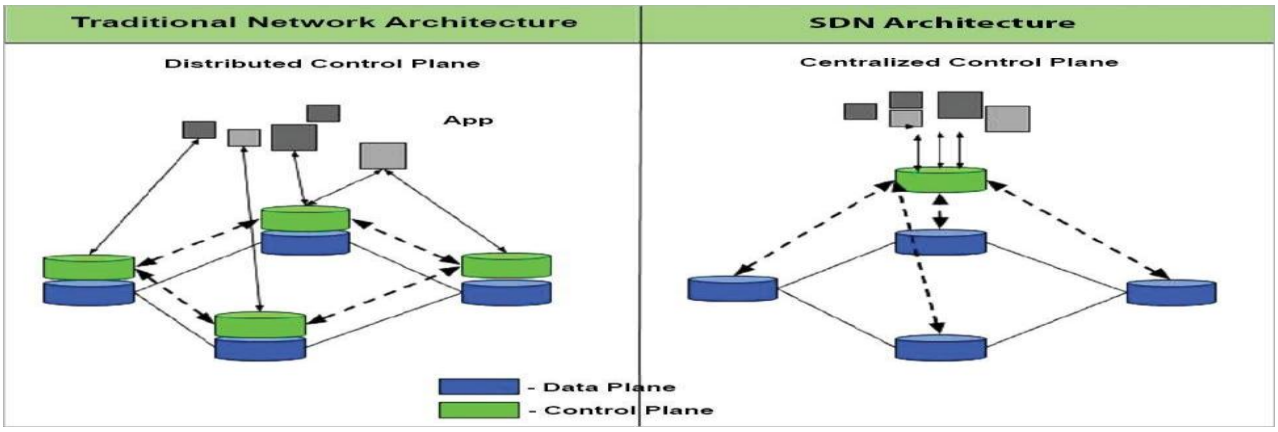

Figure 1. Traditional network vs Software-defined network architectures.

Also, software-defined networking has also a different architecture from the legacy network's architecture, which can be simply categorized or classified into 3 planes, layers or loops and they are:

- Application plane: which is represented by a whole suite of applications that are used by the admin to configure the network and enforce policies as per the business needs. So, in other words there will be direct interaction between the user or network admin with this layer. This layer is connected with the control layer via various APIs.
- Control plane: represented by the controller or group of controllers, which is the main brain of the network and these controllers could be software-based or hardware-based. It is the middle layer that connects the application layer with the data plane layer and translated the human readable high-level instructions in the application layer to machine-understandable low-level

commands in the data plane layer to achieve the intended purpose. The control plane uses communication protocols like openflow protocol to communicate with the below or (southern direction) plane as an expression for lower level which is in the data plane, in this case and north direction is for higher levels or loops.

- Data plane: it's like the muscles of the system, it is represented by the switches which are merely data forwarding devices. They don't have dedicated brain and they only do what the controller commands them to do by blocking a data flow dropping a specific packet or permitting one. The switches of this layer are also either hardware-based or software-based.

Related work:

SDN is a great technology that could affect every other technology around in a positive way, in an interchangeable way we can use other technologies to develop SDN and use SDN to enhance other technologies for example the researchers A. Wani and S. Revathi in their research [1] have used SDN-based Intrusion detection system IDS to detect threats compromising IoT networks. Others

Used SDN to create a DDoS and Dos attacks detection and mitigation methods [**2]**, while we see some researchers used the tools and techniques residing in the cloud to defend SDN environment against timing channel attacks [3]. Also, by integrating SDN with other technologies some researchers created technologies that are based on SDN like SDN-oriented WSAN (wireless sensor and actuator network) and how they succeeded designing an interface or a protocol called WSANFlow between SDN control plane and end devices (data plane) in that kind of environment [4].

Features of software-defined networks:

Before we dive into solutions for some issues in SDN we have to get a simple comprehension of some positives and negatives of SDN. Some of the positives are:

- SDN provides a better enhancement for security through the management centralization it provides using the control plane, where the system admin can configure easily the whole network regardless of its size and enforce polices required as per business needs and that also provides flexibility in management.
- Great deal of programmability, by using network functions virtualization NFV as one of its main features and services. Also, the ability to develop management apps for the application layer to make the network behave according to enterprise's needs.
- Cost saving when building and managing the infrastructure devices for the used network. So, in order to start anything new, we should know what in it for us because it could be less than the expenditures, hence it will be useless to refurbish the network and update it so, in other words some legacy network devices could be used with SDN and that means that we don't have to leave the equipment we have but, we can use them with SDN and enhance the network environment instead.
- Abstraction for layers and infrastructure [5].
- It has inevitable need to match the growing technology and data in the world.
- SDN Provides freedom and independence from vendor-constrained software and hardware by developing our own apps for our own needs and by creating open projects like the open network foundation (ONF).

Coming to the SDN cons or negatives, we can see that they are not a lot but still we have to mention them to find some remedies for them:

- Software-defined network is simply a network that can be programmed so, it's based on programming and that could mean bugs.
- Software-defined networks provide central management for the network architecture and that's a good thing to facilitate the network configuration and enhance security but, this feature itself promotes a single point of failure for the network in case of an attack on the control plane.
- Also, in case of a group or cluster of controllers the connection between controllers is called east-west bound application programming interface (API) and there are not much about securing it so, that will be among our priorities.

Ideas for future research and proposed algorithms:

Here we'll try to give a brief description for some proposed ideas to ease or solve some of the issues of SDN; most articles and researches revolve about enhancing the security of data plane in the architecture of

SDN and as we mentioned earlier that the central point of management in SDN is an advantage and disadvantage itself and since it represents (meaning the controller or control plane) the brain of the SDN structure so, it represents a great deal of importance, that's why we'll concentrate on the control plane more than other planes.

In our proposed solutions we have 3 topologies we can use any one of them and each one will use 4 algorithms.

The topologies are:

- Topology V.1: contains a group of controllers that work as main and backup but they work in a redundant like way and that's by giving a priority number to each controller and the controller with the less priority number will be the main one and it will be controlling the whole structure and every 10 seconds a copy of the main controller's configuration will be sent to the backup controller or controllers.
- Topology V.2: in this topology the only difference will be by making the group of controllers redundant 100% where there is no priority number and the group of controllers work as a whole like one entity.
- Topology V.3: it will be a mix of both topologies so, we will have also a cluster of controllers where their will be main controllers and each one of them will have its own backup controller. We'll apply the rules of topology version 1 in the relationship between the main and backup controllers, and the 2nd topology rules will be applied in the relationship between main controllers.

The algorithms are:

- HYDRA: distributed controller system, where the backup controller takes lead directly in case of a DDoS or DoS attack there will be a botnet in the proposed framework regardless of the topology used and after blocking the IP of the attacker and the infected controller we'll counter attack the attackers resource IP while all controllers in the topology will be alerted of an attack and the a new controller will be directly elected as the main controlling entity so, the attack of the attacker even if it was able to infect the main controller, it will trigger an alert in the environment so, the attacker will now lose the opportunity to continue the attack not just because his IP will be blocked but also because the environment will have a whole new main controller that was backup previously and hence comes the HYDRA-like behavior.
- VPN: since Virtual private networking provides tunneling between two endpoints, we'll use VPN to create a point to point tunneling between every 2 controllers; because VPN provides a high level of security so, in case of an attack or attempt to break the connection or the tunnel, the connection will be disconnected directly and that will alert the whole network of an attack but, in case if the attacker was able to infiltrate the VPN tunneling then, we will use the next algorithm which is RSA.
- RSA: inside the tunneling connection we'll use the public key RSA algorithm to exchange the keys of connection to start a connection session and of course that’s beside the IPsec algorithm used in VPN.
- Blockchain: blockchain is a promising technology and most people would know it from its biggest participation in information technology which is cryptocurrency. Our proposed algorithm uses blockchain technology to create blocks of hashes between controllers to authenticate and validate the connection between controllers.

The previous topologies and algorithms will be used not just to defend controllers hence defending the SDN environment but also, to secure the connection between the controllers.

Conclusion

- SDN provides Abstraction for layers and infrastructure.
- SDN is the new era of technology and it could be used for developing other fields of technology and SDN itself enhances network security by itself and by using other technologies for that purpose.
- It is a new technology so, it's still has some security issues despite that SDN itself was developed as a kind of security solution for networks.
- Blockchain could be used as a way for securing the SDN environment.

- SDN Provides freedom and independence from vendor-constrained software and hardware by developing our own apps for our own needs and by creating open projects like the open network foundation (ONF).
- SDN provides programmability and virtual environment using NFV which is both cost and resource effective

References

1. Wani A., Revathi S., Analyzing Threats of IoT Networks Using SDN Based Intrusion Detection System (SDIoT-IDS), Crescent B. S. Abdur Rahman University, Vandalur, Chennai 600048, India, ISSUE 2018, Journal of Cryptology, 7-pages.
2. Zakaria Bawany N., A. Shamsi J., Salah K., DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions, King Fahd University of Petroleum & Minerals, ISSUE 2017, Journal of Cryptology, 17-pages.
3. Anyi Liu, Jim X. Chen, Harry Wechsler, Real-Time Timing Channel Detection in a Software-Defined Networking Virtual Environment, Department of Computer Science, Indiana University―Purdue University Fort Wayne, Fort Wayne, USA, ISSUE 2015, Journal of information security ,20-pages.
4. Burhan Al-Shaikhli A., Ceken C., Al-Hubaishi M., WSANFlow: An Interface Protocol Between SDN Controller and End Devices for SDN-Oriented WSAN, Department of Computer and Information Engineering, Institute of Natural Sciences, University of Sakarya, 54187 Sakarya, Turkey, ISSUE 2018, Journal of Cryptology, 19-pages.
5. https://www.slideshare.net/martin_casado/sdn-abstractions [online][accessed: 04.03.2019].