A centralized secure plan for detecting and mitigation incidents in hybrid SDN

Karim Zkik*, *Said* EL Hajji, and *Ghizlane* Orhanou

Laboratory of Mathematics, Computing and Applications, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

Abstract. The information technology sector has experienced phenomenal growth during recent years. To follow this development many new technologies have emerged to satisfy the expectations of businesses and customers, such as Cloud Computing, mobility, virtualization, Internet of things and big data. Traditional network cannot longer support this growth and suffers more and more in terms of misconfiguration, management and configurations complexity. Software defined network (SDN) architectures can be considered as a big revolution in the field of computer networks, because they offer a centralized control on infrastructure, services and the applications deployed which facilitate configuration and management on the network. The implementation of this type of architecture is not obvious and requires great expertise and good handling and management of network equipment. To remedy this problem the SDN architectures have evolved towards distributed and hybrid architectures. Despites the advantages of using SDN, security issues remain a real obstacle in front of the deployment of this type of architecture. The centralized architecture of this type of networks makes it vulnerable to several types of attacks and intrusions, and the implementation of security equipment generally causes a decrease in performance and increase latency.

1 Introduction

According to CISCO [1] more than 50 trillion devices will be connected by 2020, 80% enterprise applications are deployed in the cloud and more than 2 million applications available between 2014 and 2016. This expansion of information technology has given rise to several challenges such as the security of computer networks, lower bandwidth cost, and the good deployment of new technologies and the management of network performance. Unfortunately, to follow the development of these new technologies, system designers often need to modify network, updates software and orchestrate computer and network resources according to the specific requirements which can be very inconvenient and very difficult to do. So, it has become essential to develop a new network architecture that can meet the needs of users.

* Corresponding author: karim.zkik@gmail.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

Software defined networks (SDN) is a new software centric approach to networking that reduces capital and operational cost through programmatic control of network infrastructure, facilitating customization, optimization, and innovation [2-3]. This new network model offers several advantages in terms of policy driven, automation and agility which make it easy to support the adoption of all kinds of technologies, services and applications. Despite the apparent benefits of using SDN architectures, security remains one of the most disconcerting challenges and issues of its deployment [4]. To overcome these problems several studies proposed solutions tailored to SDN environments [5, 6]. Unfortunately, the uses of these solutions offer somewhat limited network usability, high configuration complexity and performance consumption and do not prevent from unknown infections and zero day attacks.

In this paper we propose architecture for securing logical distributed hybrid SDN architectures. To do so, we implement a modular security plan composed from a firewall module and an anomaly detection module. By using this security plan we intend to secure SDN environment, ensure High availability, ensure threat mitigation and prevent unknown exploitation and zero day attacks.

The rest of paper is structured as follows. In section 2 we discuss some preliminaries and related works. In section 3 we present our proposed model, and we detail the conception of each part. In section 4 we present an implementation and experimental results of our work. In section 5 we present a security analysis and an extended discussion on the results. In section 6 we conclude the paper and discuss some of our future research directions for security in SDN.

2 Preliminaries and background

In this section we will define several theoretical concepts related to our work. So, we will talk about the development of SDN and we will discuss his different architecture designs. We will talk also about security issues in SDN and present the main related work to that field.

2.1 Hybrid distributed SDN architectures

SDN is a new software centric approach to networking that reduces capital and operational cost through programmatic control of network infrastructure, facilitating customization, optimization, and innovation. To deploy the SDN architecture it is necessary to replace all the current network architecture by SDN nodes which it's very difficult. So to remedy this problem researches [7] proposed the use of hybrid SDN architecture.

Hybrid deployment of SDN provides an environment where both legacy network architecture and SDN nodes can work together and which can enforce the benefits of both the traditional networks and SDN paradigm. In other hand, when using SDN architectures, we have generally a single physically centralized controller. This poses a real security problem because this controller is considered as a single point of failure. To bypass this problem, experts suggest using an SDN distributed designs [8].

Despite the advantages of using hybrid distributed SDN architecture, there is a lot of challenge that slow down the deployment of SDN [9, 10] such as: Controller-switch communication, traffic engineering, configuration, topology discovery, fault tolerance, scalability and security The security issue remains one of the most critical problems in SDN development. In this paper we will try to propose solution to overcome these challenges while focusing on security issues.

2.2 Security issues in SDN: Challenge and related works

In SDN architecture, there are several security issues that affect its operation:

- Controllers are considered as points of failure, which means that if an attacker manages to control a controller, he can control the entire network or a significant part of the network.
- When using SDN architectures, we remove the intelligent and decisional part of the SDN nodes which makes them vulnerable.
- When using Hybrid architecture, we also inherit traditional network security problems which multiply vulnerabilities and makes security operations even more complex.

We can divide the security problems in SDN architectures into 6 categories:

- Security of controllers: At first, we must secure each controller in our information system, correct flaws and vulnerabilities and detect each attempt of intrusion to these controllers.
- Security of communication between controllers: In this part it is imperative to secure the communications between controllers and to prevent attempts to listen on the network.
- Security of data Plan: In hybrid SDN, we must secure the SDN nodes and prevent intrusion attempts from traditional network architectures.
- Security of Management Plan: Securing when deploying applications and services from plan management to end users is very important. An attacker can infect applications, change the orchestration and security policies, and affect the entire network topology. Thus it is imperative to develop mechanisms for access control, intrusion management, data loss prevention and network security.
- Security of Communication: this means that you have to secure communications between the different SDN layers. This will prevent an attacker from usurping identity attacks.

To overcome this security issue several research projects have proposed solutions and secure frameworks. Table 1 lists the research work that aims to secure SDN architectures.

Table 1. An overview on related works on SDN security.

References	Contributions
Zhu and al. **2017 [11]**	SFA: A stateful forwarding abstraction processor in SDN data plane. The main goal of this work is to extend SDN controller functionality and filter malicious connections.
Xiao feng Qiu and al. **2017 [12]**	GFT: This mechanism permits also to provide security appliances with information about paths of all the flows in SDN and have a global view on the network.
Saksit Jantila and al. **2016 [13]**	Propose a new design of SDN architecture to prevent DDoS attacks. In addition, authors put a mechanism to secure data plan based on a client's access behavior.
Liyanage and al. **2015 [14]**	HIP: A protocol for securing communications between SDNs nodes and the controller. This model does not rely on IP addresses to authenticate the source of packets, but on public key usage.
Lara and al. **2014 [15]**	OpenSEC: A security model that allows making deep packet inspection, intrusion detection and malware detection.
Shin and al. **2013 [16]**	AVANT-GUARD: A secure architecture which reduces interactions between the control and the data plans and detect changing flow dynamics on the data plane. The purpose of this framework is to protect communication channels from infiltrations and prevent DoS attacks.

These framework offers somehow a complex security solution which decrease performance and increase latency on the network and existing research focuses on well knowing security threats instead of unknown and zero-day attacks.

3 A Centralized Secure Design for hybrid distributed SDN architecture

In this work we propose a new security model that proposes a solution to these issues. Thus, the main goal of our model is to secure all layer under a distributed and hybrid SDN architecture.

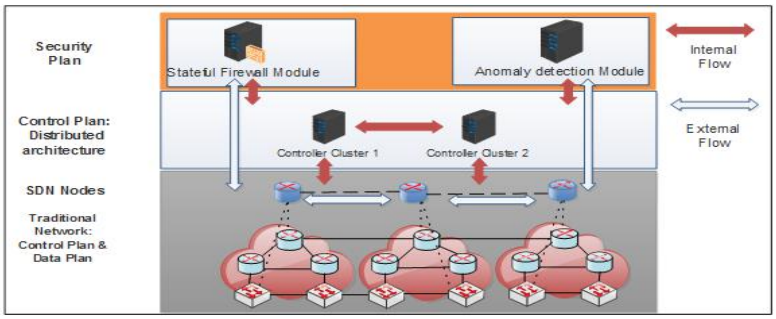

Fig. 1. Global view of proposed model.

As shown in figure 1, we propose in our architecture the addition of a new security plan. The security plan that we have integrated consists essentially of 3 modules:

- Anomaly detection Module: In this module we implement a honey controller, a DDoS detector and an analysis behavior module to detect and anomaly or security incidents.
- NIPS Server Module: In this module we propose the use of a network intrusion detection server adapted to an SDN environment to inspect the transiting packets in the network.
- State full Firewall Module: In this module we propose the use of a state full firewall adapted to an SDN environment to filter the transiting packets in the network.

3.1 Anomaly detection Module

Most attackers aim to infect SDN architectures. To do this they use sophisticated attacks and persistent malware to gain control of the network. To remedy this problem, we propose the implementation of an anomaly detection module. As shown in algorithm 1, the procedure for detecting abnormal flows and the mitigation of incident through this module is as follows:

Algorithm 1. Anomaly Detection

```
Anomaly_detector (arg1, arg2, arg3, arg4){
foreach(arg1 : Src_ip){
for(int[ ]i=0; i<T.length; i++){
if(T[i]==Src_IP){
            contain=true;}}
            else{
            alert();}
foreach(arg2 : connexion){
port_map[ stat.getDstPort() ] += 1
counts += 1}
```

```
foreach(arg3 : port_map){
incident_prob = port_map[port] / counts
l_prob = l_port_map[port]
anomaly_score += -log2[prob/l_prob]}
if(connexion_max_rate < anomaly_score){
        alert();}
foreach(arg 4: stats){
bytes += stat.getByteCount()
counts += stat.getPacketCount()}
bytes_per_second = bytes/time_interval
packets_per_second = counts/time_interval
if(packets_max_rate < bytes_per_second < packets_per_second  ){
        alert();} }
```

1- Firstly we will save the connection information of all the SDNs nodes.

2- These records will allow us to define a basic pattern of normal behavior.

3- We will make then a periodic analysis on several samples of data from the different layers.

4- An alert will be lunched if any significant deviation from this pre-established basic model was detected. In our model we verify three argument to detect anomalies:

- Firstly we will verify IP addresses and defines if the source of packets is legitimate. If the source IP does not correspond to our policies an alert will be generated.
- Secondly we will verify the destination port and defines if the connexion is legitimate and calculate the number of connexion request to detect and prevent scan attempts. To do this we used the anomaly score [17]. The anomaly score is a value that specifies the extent of the deviation of the received request compared to normal behaviour.

$$Anomaly\_Score = 0.3 \times AS_{basic} + 0.3 \times AS_{lengh} + 0.4 \times AS_{\text{threshold}} \quad (1)$$

If the destination port does not correspond to our policies, or if the connexion attempts exceeds the pre-established threshold an alert will be generated.

- Finally we will define a flow rate to prevent DoS and DDoS attack. If the received packets rate exceeds the pre-established threshold an alert will be generated.

5- The alerts number defines the probability of an anomaly on the network

6- Once an incident alert is generated, the stream is immediately redirected to management Plan.

7- A security administrator observes the infected packets and determines the nature and provenance of the security threat.

8- The administrator should take a correction action to mitigate the security incident

9- A report is generated and submitted to the management layer for further analysis.

3.2 Statefull firewall Module

In an SDN environment it is very important to use firewalls to filter and inspect incoming and outgoing packets from our information system. To do so, most SDN architecture uses stateless firewall. The choice of this type of firewall is due to the fact that it does not impact the performance. But several attacks can easily bypass this kind of equipment. Using a statefull firewall will increase the level of security but it will greatly affect performance and increase latency.

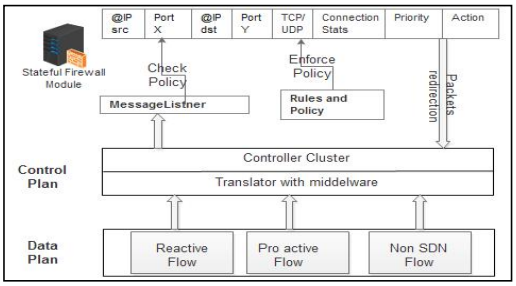

Fig.2. Adaptive Statefull firewall module for hybrid distributed SDN.

As shown in figure 2, we have developed a new design of a statefull firewall adapted to distributed and hybrid SDN environment. Connections that we will filter are coming from reactive flow, proactive flow and Non SDN flows. So, the firewall must simultaneously handle packets from SDNs and Standard network devices. To do this, we use translators at the level of data plan that translate traditional packet to open flow protocol packets. The elements of our firewall are as follows:

- **MessageListner** which receives the packets sent from control plan and compares them with the entries in the StateTable
- **StateTable** which contain the connections entry
- **Rules and policy handler** which is responsible of defining the packet filtering rule.

This module makes it possible, among other things, to filter and inspect the packets transiting the network. The centralized design of this module allows it to have visibility across the entire network and it does not affect network performance.

4 Implementation and experiment results

4.1 Experimental setup

In order to implement our AM-Sec model we have constructed a distributed SDN environment testbed using a PC Server HP DL380G6 with the following configuration: Processor Xeon quad-core E5504 2.00GHz, 4-core4MB , 80W, Memory 24GB. We use virtualization to implements controllers using an Open source virtualization platform XEN server [18].

We implement two Open Daylight controller cluster on Linux operating system with the following configuration: Ubuntu 14.02 64bits, Memory 2 GB, Processor Xeon quad-core E5504 2.00GHz

We construct a network hierarchy model by devising our topology into 2 area using Mininet Emulator [19] version 2.2.1 on Linux operating system Ubuntu 14.02 64 bits with 2 GB of RAM.

To perform a pentesting on our SDN environment we use a Linux operating system with the following configuration: Kali Linux 2017.2 64 bits, Memory 4 GB , Core i7-2670 CPU 2.20 GHz.

We use H3 ping to perform DoS and DDoS attacks. We use Ettercap to perform MitM attack.

We use Metasploit, Meterpreter and Shelter to infect SDN network. To filter and inspect packet on our network, we use Netfilter and Snort.

4.2 Implementation and experimental results

In this part we will implement the security measures and modules that we proposed and discuss in this work. For this we have launched several types of attacks on an open source and virtualized SDN environment.

We have launched several types of attacks (Dynamics Tunneling Attack, Spoofing Attack, Malware attack) at the level of control and data plans. First, it should be noted that the attacker's goal is to access and infect the server within the network. So, we have configured our firewall in advance to ban any external connection to the server. Table 2 shows the results of our implementation.

Table 2. An overview on related works on SDN security.

Security Plan Modules	Flows	Flux analysis (packets/s)	Incident Detection	Incident Analysis	Action	Latency (ms)	Controller performance (%)
Statefull Firewall modules	Normal Flow	100	-	-	Allow	0.12	9.02%
		150	-	-	Allow	0.48	9.54%
		200	-	-	Allow	0.96	10.15%
	Infected Flow	100	42	224	Drop	2.03	10.12%
		150	97	375	Drop	6.47	12.06%
		200	142	438	Drop	8.13	12.94%
Anomaly detection modules	Normal Flow	100	-	-	Allow	0.12	8.05%
		150	-	-	Allow	0.48	9.94%
		200	-	-	Allow	0.96	10.75%
	Infected Flow	100	321	-	Alert	2.03	16.49%
		150	412	-	Alert	6.47	18.73%
		200	675	-	Alert	8.13	23.81%

The results of the test demonstrate in the first place that our security plan is very reactive and that it was able to detect and stop all kinds of infections even in the case of a dynamic flow tunnelling attacks which change the flows rule to bypass firewalls. We also did several tests by changing the number of coaction requests, figure 3 and by increasing data rate, figure 4, to demonstrate the robustness of our framework. On the other hand the test results show that the use of the modular secure plan does not affect the performance of the controllers.

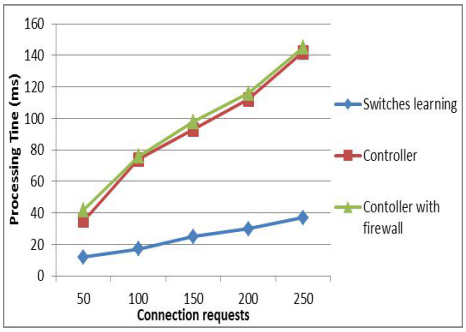

Fig. 3. Shows the behaviour of controllers and SDNs odes during the use and the absence of the modular security plan. We note that the time required for the processing is correct and that the detection and mitigation of incidents does not impact the performance of the controllers.

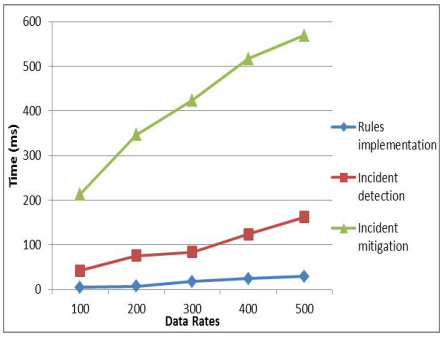

Fig. 4. Models the behavior of the firewall and NIPS modules during a security incident. The results demonstrate the robustness of our module and that the time required for the detection and mitigation of attacks remains very acceptable even at a very high flow rate.

5 Conclusion

In this paper we propose a security framework in Hybrid distributed Software defined networks environment security is one of the major concerns of experts when deploying SDN architectures. Thus, the various researches try to propose security solutions which aim at offering a good level of security without impacting the performances or increasing the complexity of the operations. So, we implement a centralized modular security plan to detect and mitigate different threats and security incident in SDN environment. Regarding our future work, we project to make our security framework more efficient and more performant and we would like to develop a model that could integrate more security models to increase safety in SDN environment. In our testbed, we used a simple virtual SDN architecture. So, it is expected to develop more suitable model adapted to a real deployment of SDN.

Acknowledgments

The author would like to thank everyone who has contributed to the progress of this research.

References

1. Evans D 2011 *The Internet of Things How the Next Evolution of the Internet Is Changing Everything* Cisco Internet Business Solutions Group (IBSG)
2. Masoudi R, Ghaffari A 2016 *Software defined networks: A survey* Journal of Network and Computer Applications
3. Nadeau T D, Gray K 2013 *SDN: Software defined networks* (O'REILY)
4. Ahmad I, Namaly S, Ylianttilaz M and Gurtov A 2015 *Security in Software Defined Networks: A Survey* IEEE Communications Surveys and Tutorials, pp 2317-2346.
5. Yoon C, Park T, Lee S, Kang H, Shin S, Zhang Z 2015 *Enabling security functions with SDN: A feasibility stud*y Computer Networks
6. Zkik k, Tachihante T, Orhanou G, El Hajji S 2016 *A Modular Secure Framework based on* SDMN for mobile core cloud Springer International Publishing, International Conference on Mobile, Secure and Programmable Networking, pp 137-152.
7. Blial O, Ben Mamoun M and Benaini R 2016 *An Overview on SDN Architectures with Multiple Controllers* Volume **2016**, Article ID 9396525, Journal of Computer Networks and Communications (Hindawi Publishing Corporation) p 8
8. Tootoonchian A and Ganjali Y 2010 *HyperFlow: a distributed control plane for OpenFlow* Proceedings of the Internet Network Management Conference on Research on Enterprise Networking (INM/WREN '10), Berkeley, Calif, USA
9. Sandhya, Sinha Y and Haribabu K 2013 *A Survey: Hybrid SDN* Journal of Network and Computer Applications
10. Benamrane F, Ben Mamoun M and Benaini R 2017 *New method for controller-to-controller communication in distributed SDN architecture* Vol **19**, International Journal of Communication Networks and Distributed Systems , pp 357-367
11. Zhu S, Bi J and Sun C 2014 *SFA: Stateful Forwarding Abstraction in SDN Data Plane* Proceedings of the ONS Research Track
12. Qiu X, Zhang K and Ren Q 2017 Global Flow Table: A convincing mechanism for security *operations in SDN* Volume **120**, Computer Networks , pp 56-70

13. Jantila S and Chaipah K 2016 *A Security Analysis of a Hybrid Mechanism to Defend DDoS Attacks in SDN* Volume **86**, Procedia Computer Science, pp 437-440
14. Liyanage M, Ahmad I, lianttila Y, Llorente J S, Kantola R, Perez O L, Itzazelaia M U, Valtierra A and Jimenez C 2015 *Security for Future Software Defined Mobile Networks* 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies
15. Lara A and Ramamurthy B 2014 Ope*nSec: A Framework for Implementing Security Policies using OpenFlow* CSE Conference and Workshop Papers, pp 781-786
16. Shin S, Yegneswaran V, Porras P and Gu G 2013 *AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks* Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp 413-424
17. Krügel C, Toth T and Kirda E 2002 *Service specific anomaly detection for network intrusion detection* Proceedings of the 2002 ACM Symposium on Applied Computing, SAC '02, ACM, New York, NY, USA, pp 201–208
18. *XenServer 6.x Best Practices* 2013 Dell Compellent Storage Center
19. *Introduction to Mininet*, https://github.com/mininet/mininet/wiki/Introduction-to-Mininet