2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)

Framework for Securing SDN Southbound communication

Bhushan Pandya
SVKM's NMIMS,
MPSTME,
Mumbai
bhushan132@gmail.com

Sonal Parmar
SVKM's NMIMS,
MPSTME,
Mumbai
sonal.parmar@nmims.edu

Dr. Zia Saquib
Centre for Development
of Advanced Computing
(CDAC), Mumbai
saquib@cdac.in

Anupam Saxena
Centre for Development
of Advanced Computing
(CDAC), Mumbai
anupam.saxena@cdac.in

***Abstract*—The programmable separation of control and forwarding elements of networking which allows programmability and open innovation and separation of basic hardware from various control and management functions is the work of Software Defined Networking (SDN). With the major role of equipment controllers and the reality of security threats, it is also important to leverage defense-in-depth, protocol security, and a reduction in the controller's attack surface to make sure that the SDN is secure and is operating properly. This paper suggest a framework for security services using two representative security systems, such as with security middlebox such as IDS (Intrusion Detection System) and cryptographic framework using digitally signed Message which addresses data integrity and authorization. This paper discusses the limitations of existing systems and presents a possible SDN-based security framework to protect network resources by controlling and prevention of suspicious and dangerous threats.**

Keywords- SDN; Southbound; Security; Integrity; Authorization; Cryptography.

I. Introduction To SDN

SDN is widely accepted as a new paradigm in network designing and components. It has enabled an open framework to create modular networks that can be functionally separated, controllability, manageability, application and service awareness. SDN’s key characteristics are: Modularization of Hardware, Software and Functions, Multi-Tenancy, Centralization of Control, Multi-Level Virtualization, Programmability and Open Innovation. An SDN architecture can be depicted as a composition of different layers and has its own specific functions, as shown (Figure1). A SDN deployment comprises of the southbound application programming interface (API), northbound API, network operating systems and applications. Southbound APIs are the connecting bridges between control and forwarding elements, thus being the crucial instrument for clearly separating control and data plane functionality [1].

OVSDB (Open Virtual Switch Data Base) [16] is type of southbound API, acts as a complementary protocol to OpenFlow [17] for OVS (Open Virtual Switches) which is designed to provide advanced management capabilities. OpenFlow’s capabilities are not just limited only for the configuration and behavior of flows in a forwarding device, it offers other networking functions, it allows the control elements to configure tunnel interfaces on OpenFlow data paths, set QoS policies on interfaces, create multiple virtual switch instances, attach interfaces to the switches, manage

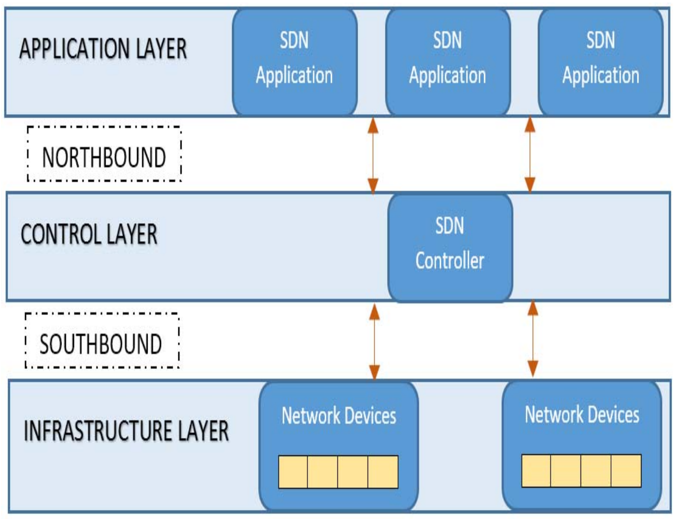

Fig. 1: SDN Architecture

queues and collect statistics. Traditional networking systems enable abstractions for accessing lower-level devices, manage the parallel access to the underlying resources and security protection mechanisms [3].These functionalities and resources are key enablers for increased productivity, making efficient system and application. SDN is promised to facilitate network management and ease the burden of solving networking problems by single entity that manages all forwarding devices of the network by means of the centralized controller. However, it represents a single point of failure and may have scaling limitations, it may not be enough to manage a network with a large number of data plane elements.

II. Related Work

The identified threats vector in SDN architectures which consists of forged or faked traffic flows in the data plane, which can be used to attack forwarding devices and controllers that allows an attacker to exploit vulnerabilities of forwarding devices and consequently wreak havoc with the network as discussed in [2]. By attacks on the control plane and data plane can easily grant an attacker the control of the network. The other vector is linked to attacks vulnerabilities in administrative stations, faulty or malicious controller or application could be used to reconfigure the entire network for data theft purposes, as in a data center. Threat vectors specific to SDN as they stem from the separation of the control and data planes and the consequent introduction of a new entity in these networks – the logically centralized controller [4].A compromised critical computer, directly connected to the control network, will empower the attacker with resources to launch more easily an attack to the controller. Distributed denial of service (DDoS) [2] attacks in cloud computing environments are growing due to the essential characteristics of cloud computing. With recent advances in SDN, brings us new chances to defeat DDoS attacks in cloud computing environments. The capabilities of SDN which includes global view of the network, dynamic updating of forwarding rules, centralized control, software-based traffic analysis, makes it easier to sense and response to DDoS attacks. However, the security of SDN itself remains to be addressed, and potential DDoS vulnerabilities exist across SDN platforms, as shown in (figure 2). However SDN-based cloud shows many good characteristic, it faces several challenges that must be taken into consideration, including performance, availability, scalability and security.

Performance: the processing speed of the network node considering both throughput and latency [5].The process of SDN to handle new packets brings the programmability and at the same time it leads to performance problems [6].

• Availability: it refers to the proportion of time an SDN system is in a functioning condition, the dependency on the controller creates obstacle. One advantage of a traditional over SDN, distributed network architecture is that if a switch fails, the availability of the network can be maintained [7].

• Scalability: the ability to be extensive to accommodate network enlargement. By introducing distributed or peer-to-peer controller infrastructure may share the communication burden of the controller. But an overall network view is required to direct the communications between the controllers, besides there are some other scalability concerns including the flow setup overhead and resilience to failures [9].

• Security: SDN and attacks have a contradictory relationship. Security analysis has showed that the SDN framework suffers many security threats, including : 1) data leakage, e.g., flow rule discovery (side channel attack on input buffer) forwarding policy discovery (packet processing timing analysis), 2) malicious applications, e.g., fraudulent rule insertion controller hijacking, 3) unauthorized access, e.g., unauthorized controller access or unauthenticated application access, 4) data modification, e.g., flow rule modification to modify packets, 5) configuration issues, e.g., lack of TLS (or other authentication techniques) adoption policy.

Several recent efforts have been made in order to address various security concerns, such as vulnerability assessment, DDoS attack detection and saturation attack mitigation, in SDNs. FortNOX [11] was proposed as a software extension aiming to provide security constraint enforcement for OpenFlow controllers, being able to identify indirect security violations. On one hand, the rule conflict analysis algorithm provided by FortNOX records rule relations in alias sets, which are unable to accurately track network traffic flows as the conflict detection algorithm in FortNOX only conducts pairwise conflict analysis between new flow rule(s) and each single security constraint without considering rule dependencies within flow tables and among security constraints, when FortNOX detects a security violation caused by new rule(s) installed by a non-security application, it simply rejects the rule(s) without offering a fine-grained violation resolution. [12]. Apart from FortNOXs numerous firewall algorithms and tools have been designed to assist system administrators in managing and analyzing firewall policy anomalies like VeriFlow [13] and NetPlumber [14].

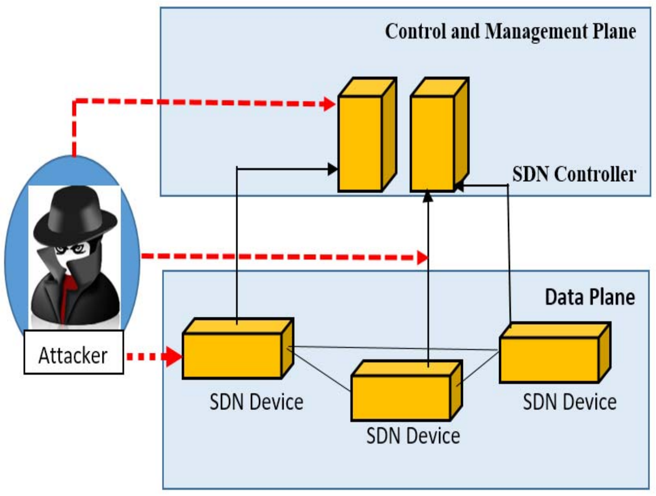

Fig. 2: SDN Threat Attacks

Even though these tools can be potentially used to detect attacks, they could not support automatic and effective resolution. Also, they overlook the other security challenges and ignore securing the communication channel within. In addition, they are unable to check state full network properties. However, existing security tools only detect policy anomalies within a firewall policy, but cannot be directly applied to deal with firewall policy violations against flow policies in dynamic OpenFlow networks with respect to network-wide access control. Cyber-attacks against financial institutions, energy facilities, government units and research institutions are becoming one of the top concerns of governments and agencies around the globe. As expected, one of the most common means of executing those attacks is through the network, either the Internet or the local area network Industry experts believe that security and dependability are issues that need to be addressed and further investigated in SDN. Differentiating from those work, and our work focuses on exploring how to secure the southbound communication for SDN environment by securing the communication channel and by observing the flow of the traffic throughout the network.

III. Proposed Framework

Different threat vectors have already been identified in SDN architectures, as well as several security issues and weaknesses in OpenFlow-based networks [2]. While some threat vectors are common to existing networks, others are more specific to SDN, such as attacks on control plane communication and logically-centralized controllers. It is noticeable that most threats vectors are independent of the technology or the protocol because they represent threats on conceptual and architectural layers of SDN itself. Southbound APIs facilitate efficient control over the network and enable the SDN Controller to dynamically make changes according to real-time demands and needs. The ability to use APIs to create more user-friendly management interfaces increases the attack surface of the network infrastructure considerably, because security is no longer limited to the network equipment. Potential attackers can try to get control of the network, pretending to be an SDN controller or just break into the controller itself. Access to the SDN control network should be controlled to prevent unauthorized activity. As the network's centralized decision point, it is first critical to secure the SDN controller within the architecture. Strong access control is required, hence the communication channel between the Controller and the OVS should be considered and secured. An SDN architecture along with the proposed security framework to secure the southbound API is shown in (figure 3).

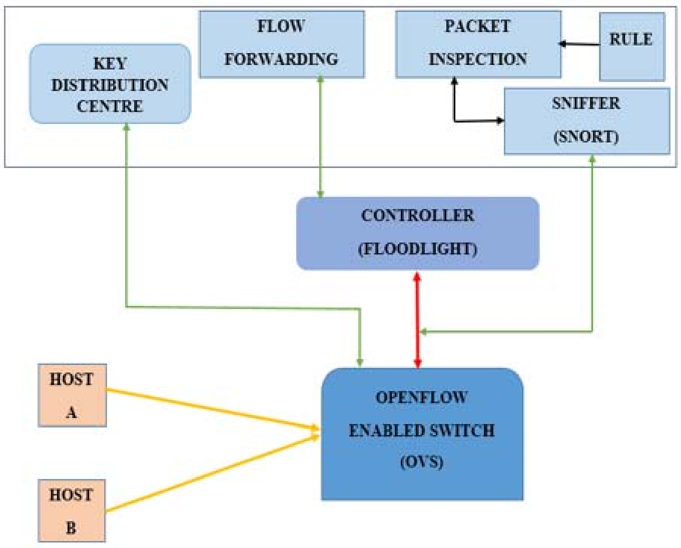

Fig. 3:Proposed Security Framework

We overcome the challenges by deploying an Intrusion Detection Centre (IDC), Key Distribution Centre (KDC) between control plane and data plane which not only monitors the traffic in southbound but also authenticate and secure the communication link in-between the control and data plane. A middle-box solution checks for securing the centralized controller and check for the authorized device access only. As discussed in earlier chapter in order to secure a SDN environment we need to secure the southbound as well as northbound devices and the communication link in-between.

An Intrusion Detection system by using Snort and Kerberos Key distribution center authenticates as well as monitor the newly and previous connected switches (OVS, Virtual switch and routers) in-between the southbound devices. Snort which acts an IDS sniffs all the traffic flowing through the communication link between data and control plane devices while the Kerberos KDC verifies the authenticity for devices connected to the southbound environment. The promiscuous mode is enabled it passes every packet to the administrative device as shown (figure 4), is a feature of a network interface card. Ordinarily, a NIC passes only the packets actually destined to its host machine rest are discarded, this allows the host to spy on all packets on the network. The management network handles communication among nodes. The data network handles communication coming to and from (Virtual Machines) VMs. The external NIC connects the network node, and optionally to the controller node, so your VMs can connect to the outside world. The configured promiscuous mode on a VMware Virtual NIC, the Virtual Switch sends a copy of every packet received by the Virtual Switch to that Virtual NIC.

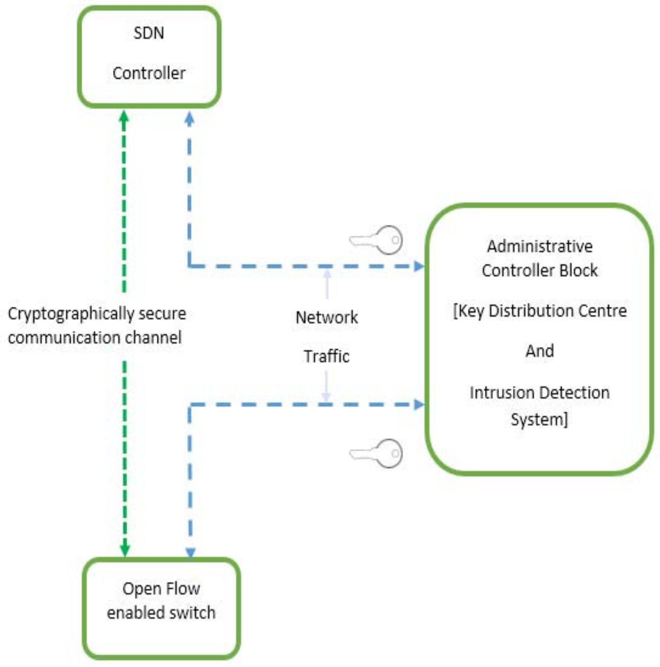
SDN Controller
Administrative Controller Block [Key Distribution Centre And Intrusion Detection System]
Cryptographically secure communication channel
Network Traffic
Open Flow enabled switch

Fig. 4: Proposed Framework overview

Rather than getting a few stray packets for which the switch does not yet know the correct destination, the Virtual NIC gets every packets. Snort sniffs the traffic and the activity log file will be generated and stored at the administrative device, which can be monitored for any unusual activities. A Kerberos KDC is also implemented to authenticate all the devices which are connected to Southbound. The Administrative device maintains the generation assignment and database of keys of each existing and new nodes to which it wants to communicate and if key is not available in its database then it requests to Administrative for obtaining required Key before initiation of communication. All keys and Keys are confidential and secured and stored in hash value, any node can request to Administrative for getting Key of other node only if it wants to communicate. After initiation request for communication both party (sender and receiver) gets assigned with the Keys and the same are stored in the database of Administrative system. In general Administrative system authenticates any node on the network by issuing certificate to that node but in this case extra computational overhead of certificate might be an issue for some nodes. In such cases we have two options, first one is to reduce certificate as much as possible by removing unnecessary extra fields from it [18], and second is to replace certificate value with a single unique value like "MAC Address" of the corresponding entity. To lower the memory usage and make the framework lightweight it can store a limited number of counter keys in their databases. This limited storage of keys can cause an extra overhead at run time, if required key is not available in database. To overcome this problem, scheme uses 'dynamic arranged database for optimal key storage'. Each node stores Key of Administrative device in first row of database, always new Key will be stored in next row of database and all keys will shift downward by one row. Key at the bottom row is removed if database is already full. If any CounterKey is used by the node from its database then this used key will be shifted to third row and all Keys (which were above in database from used counter key) will shifted downward by one row. The place of Keys those are at lower position from used counter keys will be unchanged.

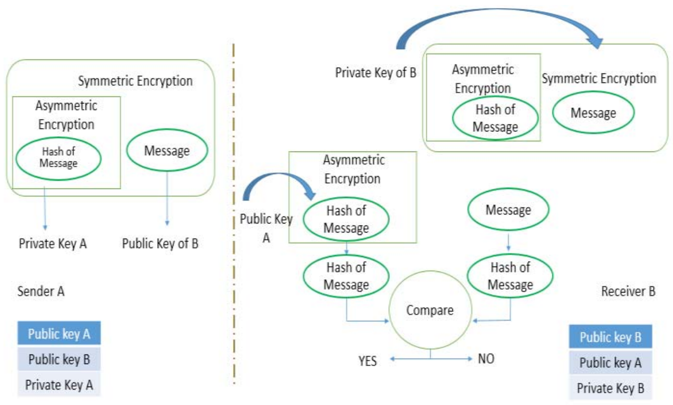

Fig.5: Securing the link using Digital Signature

However we aim to secure the southbound communication link by monitoring the traffic as well as check for the authenticated devices [15].The southbound interface between the controller and data-forwarding devices is vulnerable to threats that could degrade the availability, performance and integrity of the network. The Southbound APIs main function is to enable communication between the SDN controller and the network nodes (both physical and virtual switches and routers) so that the router can discover network topology, define network flows and implement requests relayed to it via northbound APIs. The proposed framework secures the communication link between Controller and the Switch by cryptography. The Message which are send across the Control plane and Data plane are digitally signed and encrypted by symmetric key encryption. The hash of the same message is encrypted using asymmetric encryption and this is attached with the message itself which is then encrypted using symmetric key encryption.

This encrypted data is then sent across the network, at the receiver's end which contains the public key of the sender decrypts the message thus obtaining the hash of the message and the message. Hash value of the message is then obtained, to ensure the integrity of the received data, hash of the message which is received is compared with the received pre-defined hash value as shown in Figure 5. If the hash value of both the messages matches the integrity of the message is prevailed. If the hash value is not matched then the data received is discarded. Thus the proposed security framework focuses on achieving authorization and prevailing the message integrity among the southbound communication. The main focus is on keeping the proposed framework as lightweight as possible, also as to make is as secure and reliable.

ACKNOWLEDGMENT

First and foremost, I would like to express my deep sense of gratitude and indebtedness to my Parents and supervisor Assistant Professor Sonal Parmar for their invaluable encouragement, suggestions and support from an early stage of this project and providing me extraordinary experiences throughout the work. I am highly grateful to Dr. Zia Saquib, Executive Director, CDAC, Mumbai and Mr. Anupam Saxena, CDAC, Mumbai for their kind support and permission to use the facilities available at the Center for Development of Advanced Computing (CDAC), Mumbai. Above all, there priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

I am obliged to Prof. (Dr.) Vaishali Kulkarni, HOD-EXTC, and Dean Dr. Sharad Mhaiskar, MPSTME, NMIMS University for giving me an opportunity to present a project in the field of Network Security.

REFERENCES

[1] C. Janz, L. Ong, K. Sethuraman and V. Shukla, "Emerging transport SDN architecture and use cases," in *IEEE Communications Magazine*, vol. 54, no. 10, pp. 116-121, October 2016.
[2] D. Kreutz, F. M. V. Ramos, P. E. Ver?ssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015
[3] W. Xia, Y. Wen, C. H. Foh, D. Niyato and H. Xie, "A Survey on Software-Defined Networking," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27-51, first quarter 2015.
[4] Qiao Yan, F. Richard Yu, Qingxiang Gong and Jia nqiang Li," Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," in IEEE Communications Surveys & Tutorials, VOL. 18, no. 1, pp.602-622 ,First quarter 2016
[5] S. Sezer et al., "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communication Mag., vol. 51, no. 7, pp. 36–43, Jul. 2013.

[6] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Communication Surveys Tuts., vol. 15, no. 2, pp. 843–859, 2nd Quart. 2013.

[7] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," IEEE Communication Surveys Tuts., vol. 15, no. 4, pp. 2046–2069, 4th Quart. 2013.

[8] S. Sezer et al., "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communication Mag., vol. 51, no. 7, pp. 36–43, Jul. 2013.

[9] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," IEEE Communication Mag., vol. 51, no. 2, pp. 136–141, Feb. 2013.

[10] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in Proc. IEEE SDN4FNS, 2013, pp. 1–7.

[11] P. Porras et al., "A security enforcement kernel for openflow networks," in Proc. 1st Workshop Hot Topics Software Defined Network, 2012, pp. 121–126.

[12] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su, P. Mohapatra and C. Davis, " Fireman: A toolkit for firewall modelling and analysis," in IEEE Symposium on Security and Privacy, 2006.

[13] P. Kazemian, G. Varghese, and N. McKeown. Header space analysis: static checking for networks. In NSDI'12.

[14] P. Kazemian, M. Chang, H. Zeng, G. Varghese, N. McKeown, and S. Whyte. Real time network policy checking using header space analysis. In NSDI'13.

[15] Anupam Saxena, Om Pal, Zia Saquib and Dhiren Patel, " Customized PKI for SCADA System ," in Int. J. of Advanced Networking and Applications , Volume: 01, Issue: 05, Pages: 282-289 (2010).

[16] http://openvswitch.org/

[17] http://openflow.org/

[18] Ludovic Piètre-Cambacédès, Pascal Sitbon, "Cryptographic Key Management for SCADA Systems, Issues and Perspectives," proceedings of the 2008 International Conference on Information Security and Assurance (isa 2008) Pages 156-161.