# Survey on SDN Security Mechanisms

Teenu Jose
Toc H Institute of Science and technology
Arakkunnam-682313
Ernakulam, Kerala

Jincy Kurian
Toc H Institute of Science and technology
Arakkunnam-682313
Ernakulam, Kerala

## ABSTRACT
Software Defined Network is developed by the demand of large business organization. Software Defined Network (SDN) is a new network architecture that provides central control over the network. SDN achieve the centralized control by separating the network control from forwarding and is directly programmable. In SDN there is a controller which controls the entire network. The main benefit of SDN is the centralized control and network control programmability. central control is the major advantage of SDN but it leads to central point of failure. The programmability of network control leads to another vulnerabilities. However, before any large scale deployments, it is important to understand security issues arising from this new technology. This survey paper deal with SDN security issues and 9 security mechanism.

## Keywords
Software Defined Networking, DoS Attack, OpenFlow

## 1. INTRODUCTION
SDN is a new network design that separates the control plane and data plane. The control plane is separated from individual network devices and moved on a controller. In the traditional approach to networking, most network functionality is implemented in a dedicated appliance; i.e., switch, router, application delivery controller. In addition, within the dedicated appliance, most of the functionality is implemented in dedicated hardware such as an ASIC (Application Specific Integrated

Circuit). Networking organizations are under increasing pressure to be more efficient and agile than is possible with the traditional approach to networking. One source of that pressure results from the widespread adoption of server virtualization.

The Open Networking Foundation (ONF) is the group that is most associated with the development and standardization of SDN. According to the ONF, Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow protocol is a foundational element for building SDN solutions.

The characteristics of SDN architecture are Directly programmable, Network control is directly programmable because it is decoupled from forwarding functions. Agile, Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs. Centrally managed, Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch. Programmatically configured, SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software. Open standards-based and vendor-neutral, When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols. The rest of the paper is arranged as follows: Section 2 contain the background. The attacks on SDN are described in section 3. Section 4 deals with the security mechanisms.

## 2. BACKGROUND
The emergence of SDN is mainly for the business organization. As the business grows the organizations are need to work with large data centers, this require a change in the network either the network should configure to a level so that the network would satisfy the requirements or enlarge the network. The second option is very expensive so we go for the first option to dynamically change the network configuration that is actually happening in SDN.

In traditional network each device has its own controlplane and dataplane hence we can say that each device in traditional network is proprietary. Each device is controlled by itself and there is no any globalized view. The SDN achieve this centralized control by moving the controlplane into a controller called SDN controller. The SDN controller is the core part of SDN architecture. There are some SDN controller NOX, POX, FloodLight etc.

SDN have 3 layer architecture. The layers are application, control and infrastructure. The application layer contain business application. The SDN controller is situated in controller layer and the layer also contain network services like load balancing, routing etc. as already mention that the network devices are controlled by the controller software, so there is a protocol for the communication between the controller and the devices. OpenFlow is such a protocol.

The paper by Hyojoon Kim and Nick Feamster[1] propose SDN as a key for network management and network configuration. SDN separate the dataplane and controlplane. The network is controlled by the controller and the dataplane devices just forward the data packets. SDN support frequent changes to network conditions and state, providing support for network configuration in a high level language, and providing better visibility and control over tasks for performing network diagnosis and troubleshooting. SDN bring several advantages, first is through a software program change the behaviour of network devices. Second it will give a centralized control so there is a no need to configure individual devices to change the network behaviour instead the controller make the forwarding decision.

This paper also propose a method to implement high level policies in SDN called Procera. Procera is based on functional reactive programming (FRP). FRP is a programming for asynchronous dataflow programming using the building blocks of functional programming. Procera allows operators to express highlevel policies with this language, and translates such polices into a set of forwarding rules, which are used to enforce the policy on the underlying network infrastructure, using OpenFlow.

The papers[2]by Jacek Wytr˛ebowicz "SDN Controller Mechanisms for Flexible and Customized Networking" is a reference for SDN controller(SC) functionality. Controller mechanism forms the base for autonomous network operations. It is possible to deploy any application directly on SC, due to security reasons application for network management and provisioning of network devices are allowed to deploy on SC. Routing is a simple task for SC because it knows the status and all information about the network. There is an exception case, external routing it requires routing via border interfaces. In such situation an application is run on a host that take care of the external routing. B4 SDN is an example for a network that uses such an application. Other functionalities of SC are Access Control for Flows (AC4F),Multipath Unicast and Anycast Routing for QoS and Load Balancing Control (MPR),Multicast Routing for QoS Control (MCR),Path Selection for End-to-End QoS Flows (PS4Q),Transactions for Flow calendaring (T4C),Energy Savings Control (ESC),Cooperation with other SDN controllers (C2C).

## 3. ATTACKS ON SDN

The SDN attacks can be classified into 3 categories. Controlplane specific which includes attack cases against SDN control and application layer. Control channel specific which includes attacks targeting to the interface(OpenFlow). Dataplane specific which includes the attack on network devices. The important aspect of security is the availability of network resources so the main attack is DDoS attack.

There are different types of DDoS attacks on SDN[9]. Application Layer DDoS Attacks: There are two methods to launch application layer DDoS attacks: attack applications, or attack the northbound API. Since isolation of applications or resources in SDN is not well solved, DDoS attacks on one application can affect other applications.

Control Layer DDoS Attacks: The controllers could potentially be seen as a risk of single point of failure for the network, so they are a particularly attractive target for DDoS attacks in the SDN architecture. The following methods can launch control layer DDoS attacks: attacking the controller, the northbound API, the southbound API, the westbound API, or the eastbound API. For example, many conflicting flow rules from different applications may cause DDoS attacks on the control plane. Within the operation of SDN, the data plane will typically ask the control plane to obtain flow rules when the data plane sees new network packets that it does not know how to handle. There are two options for the handling of a new flow when no flow match exists in the flow table: either the complete packet or a portion of the packet header is transmitted to the controller to resolve the query. With a large volume of network traffic, sending the complete packet to the controller would occupy high bandwidth.

Infrastructure Layer DDoS Attacks: There are two methods to launch infrastructure layer DDoS attacks: attack switches or attack the southbound API. For example, if only header information is transmitted to the controller, the packet itself must be stored in node memory until the flow table entry is returned. In this case, it would be easy for an attacker to execute a DoS attack on the node by setting up a number of new and unknown flows. As the memory element of the node can be a bottleneck due to high cost, an attacker could potentially overload the switch memory (e.g. targeting to exhaust TCAMs). The generated fake flow requests can produce many useless flow rules that need to be held by the data plane, thus making it difficult for the data plane to store flow rules for normal network flows .

SDN also suffer from several attacks like Packet flooding: Packet-In message is one of the OpenFlow messages for describing each unseen flow. An attacker generates a number of distinct network flows, numerous Packet-In messages are sent to the controller, and the resources of the controlplane will be consumed Those a bunch set of Packet-In messages could make a controller be under an unpredictable state. Control message manipulation: Control message has responsibility for communicating between the control plane and the data plane. Through manipulating control messages, an attacker performs Switch Table Flooding, Switch Identification Spoofing, and Malformed Control Messageattacks. The attack sequences is an attacker manipulates control messages and controller parses sent control messages and worked unexpectedly.

## 4. SECURITY MECHANISM FOR SDN

FRESCO[3] is a security framework for SDN. It allows composition different modules. It provide an API that allow the security practitioners to program different modules such as monitoring, detection module these modules are then known as modular library. These modular libraries are combined to form a security framework. FRESCO have 16 library modules and each module has 5 interfaces, by assigning value to these interfaces and connecting several modules FRESCO can create different security framework. The main advantage of FRESCO is to develop different security framework by changing the library modules. The disadvantage is that the correctness of the security mechanism greatly depend upon the changes made not upon the FRESCO framework.

There is a tool for the analysis of openflow security called STRIDE[7]. The STRIDE uses a DFD of the target system and the DFD contain the system's component, process, data stores and data flows. . Using this DFD the analyst find out the vulnerabilities at each component. For this use the STRIDE mnemonic: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. If the analyst wants to find DoS attack on the controller, he will analyse the DFD and evaluate the effect of the attack to the overall system. The analyst result contain the system component and vulnerability pairs This paper also mention about attack trees. The attack tree illustrate the main attacks and the intermediate attacks. The root of the tree will be the main objective of the attack, in the case of controller the root contain the controller components and the vulnerability pair derived using STRIDE. Intermediate objectives of the attack is represented by sub nodes of the tree. Leaf nodes contain action or events. The analysis is start from the root node and the sub nodes are developed by decomposing the root objective. This tool is used to detect the attack, it does not provide any prevention method. This paper focuses a number of mitigation techniques but they are not proven in the work.

lightweight method for DDoS attack detection[8] the NOX registered switches are monitored in each time interval. During such intervals, features of interest flows are extracted. These extract features are converted into 6 tuple form and moved to classifier module. The classifier module identify whether this feature contribute any DDoS attacks. The method is divided into three modules. That are Flow Collector, feature extractor and classifier. Flow controller request flow entries periodically from the OF switches. This communication is happened through secure channel. The Feature Extractor will receive the flows and convert it into 6-tuple form. This 6 tuple form of data is then move to the classifier. The Classifier module analyzes and find whether it is belong to DDoS flooding attack. The classifier use statisticalor learning method to make such decision. In this work SOM classification method is used. This method also specifies the detection method not the prevention method.

There exist an authentication and access control for SDN[4]. This mechanism involve authentication using datalink layer protocols. The proposed mechanism applies IEEE 802.1X standard and Extensible Authentication Protocol (EAP). A virtual router sends an authentication request, standardized by IEEE 802.1X, and POX controller redirects it to the Authenticator. Then, the Authenticator responds it and the Supplicant host sends its credentials. The Authenticator checks the credentials of the Supplicant against RADIUS server, running the authentication method defined in EAP. If credentials are correct, the Authenticator sends a success message for Supplicant host and sends an authorization and confirmation message for POX via a SSL secure channel. This message identifies the Supplicant by its MAC address, confirms the success of the authentication, and also informs the identity of the Supplicant. After authentication, our application running over POX allows the supplicant host to access network resources. In case of revocation of the credentials of the supplicant host, the Authenticator communicates POX, which immediately denies the host access to the network. This authentication is done at the datalink layer so less overhead as compared to that application layer overhead. The disadvantage is that if an attacker uses a legitimate users credential to attack the network then this method revoke the legitimate users eventhough there is no problem with the legitimate users.

FortNox[10] is a new security policy enforcement kernel. It is an extension to the open source NOX OpenFlow controller. FortNOX use role-based authentication to determine the authorized person.

Defines 3 threat vectors that are significant to SDN and the security measures[5]. The first threat vector is attack on control plane communication. The weakest link between the controller and the device may introduce DoS attack. Once the attacker get an access to control plane then it will launch DoS attack. Several papers recommended that the TSL/SSL mechanisms are not the optimal solution. The proposed solution are diversity, dynamic switch association and trust between controller and devices. Second threat vector is attacks on and vulnerabilities in controller. This is considered as the severe threat. If an attacker is able to attack the controller then he or she can compromise the entire network. The solutions are trust between controllers and apps, security domains and secure components. The last threat vector is lack of mechanisms to ensure the trust between the controller and management application. Possible solutions are secure components, security domains and replication.

There is a method called Ethane[6] it is similar to basic SDN architecture there is a centralized controller that implement the security mechanism and the Ethane switches simply forward the data packets. The drawback of this method is that the application traffic compromise the network security. Another mechanism is FlowTag architecture that uses minimaly modified middleboxes. There is an API that is used by the middlebox to communicate with the controller. The FlowTag contain traffic flow information in packet headers that provide flow tracking and controlled routing. The disadvantage is that it works on predefined policies and does not support dynamic action.

Moving target defence mechanism(MTD)[12] is a work that protect the network from the attacks by using dynamic network configuration. The advantage of this work is that it respond to dynamic action. But it suffers from several disadvantages. The MTD introduce extra operational cost, It does not deal with future security problems. These disadvantage make this work impractical.

## 5. CONCLUSION

SDN is an emerging architecture. This architecture avoids several demerits of traditional network and meet the requirements of business application. SDN has several advantages but it suffers from security issues. This paper studied some security mechanism used in SDN. SDN is a future network architecture so it is important that to use secure SDN. SDN is developed to change the network configuration dynamically in order to meet the requirements. The main disadvantage of the above discussed mechanisms except MTD is they work with static environment. MTD is responds to dynamic environment but it have disadvantages. The static network configuration allow the attacker to find the weakness of the network and the attacker can easily attack the network for this reason a dynamic network configuration environment is needed ie SDN, but the security mechanism are based on static environment. So need a security mechanism that will respond to dynamic action. SDN is the future network architecture, so the future scope of this study is to develop a security mechanism that responds to dynamic action.

## 6. REFERENCES

[1] Hyojoon Kim and Nick Feamster: “Improving Network Management with Software Defined Networking”, February 2013

[2] Jacek Wytr¸ebowicz, Thorsten Ries, Khoa Truong Dinh, and SławomirKukli´nski: “SDN Controller Mechanisms for Flexible and Customized Networking”, September 2014

[3] Seugwon Shin, Phillip Porras, Vinod Yegneswaran, Martin Fong, GuofeiGu, Mabry Tyson: “FRESCO: Modular Composable Security Services for Software-Defined Networks”, February 2013

[4] Diogo Menezes FerrazaniMattos, Lyno Henrique Gonc¸alvesFerraz: “AuthFlow: Authentication and Access Control Mechanism for Software Defined Networking”

[5] Diego Kreutz, Fernando M. V. Ramos, and Paulo Verissimo: “Towards Secure and Dependable Software-Defined Networks”, August 2013

[6] Sandra Scott-Hayward, Gemma O’Callaghan and SakirSezer: “SDN Security: A Survey”

[7] RowanKloti, VasileiosKotronis and Paul Smith: "Openflow : security analysis"

[8] Rodrigo Braga,EdjardMota andAlexandre Passito: "Lightweight DDoS Flooding attack detection using NOX/Openflow"

[9] Qiao Yan and F. Richard Yu: "Distributed Denial of Service Attacks in Software-Defined Networking with Cloud Computing" April 2015

[10] Philip Porras, Seungwon Shin: "A security enforcement kernel for openflow networks"

[11] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks,"Communications Magazine, IEEE, vol. 51, no. 7, 2013.

[12] S.Jajodia et al. Moving Target Defense, Springer, 2011