Sixth International Conference on Emerging trends in Engineering and Technology (ICETET'16)
www.ijera.com
ISSN: 2248-9622, pp.14-19

RESEARCH ARTICLE OPEN ACCESS

Firewall A New Approach to Solve Issues in Software Define Networking

S.Vasudevan
Assistant Professor CSE Department, Chendu College Of Engineering And Technology, Chennai.

Abstract: In this Modern World Software Defined Networks (SDN) is becoming an developing technology. This technology helps to solve many network related problems in an effective way because SDN provide a centralized network control and provides direct programmability in the network plane. The Openflow Protocol is the basis of SDN which enables the decoupling of control and data plane in SDN which also creates vast problems such as unauthorized access, Distributed Denial of Service (DDoS), anomalies in the Openflow policies in Openflow-enabled switches and conflicts in Firewall Policies and handling traffic. The above problems are overcome by SDN oriented distributed firewall. This paper concentrates on how the SDN oriented distributed firewall ensures security in SDN Controllers handles traffic and detects conflict rules as well as prevents the conflict rules in Openflow Switches as well as in Firewall.

Keywords: Software Defined Networks (SDN), Firewall, Openflow Protocol, OpenFlow-Enabled Switches, Distributed Denial of Service (DDoS), SDN Controllers.

I. INTRODUCTION

To share the resources between two or more computers the information is converted to digitalized packets and pass through various routers and switches with the help of Transport Network Layer and Control Protocols. This is the basic concept of Computer Network.

Lately SDN has become one of the most popular subjects in the ICT domain. This paper is arranged in the order by Sections. Section II describes about the Definitions of SDN & Firewall. Section III covers about the various Issues occurring in SDN. Section IV focuses the solutions for the issues occurring in the SDN. Section V discusses about the Implementations. The Conclusion of this paper is discussed in Section V.

II. DEFINTIONS

A) Software Defined Networks

A software-defined network (SDN) is an emerging networking paradigm that gives hope to change the limitations of current network infrastructures. First, it breaks the vertical integration by separating the control plane from the underlying routers and switches that forward to the data plane. Second, with the separation of the control and data planes, network switches become a simple forwarding devices and the control logic is implemented in a logically centralized controller, A functional view of SDN architecture is shown in

Fig1

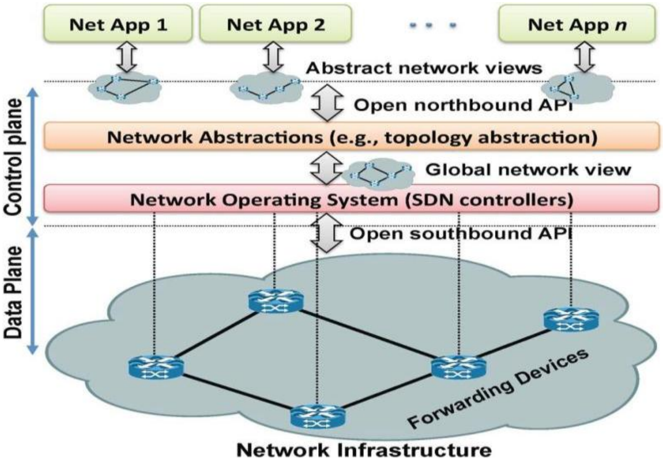


Fig: 1. Functional Architecture of SDN

Terminologies

1) Data Plane (DP): Forwarding devices are interconnected through wireless radio channels or wired cables. The network infrastructure comprises the interconnected forwarding devices, which represent the data plane.

2. Control Plane (CP): Forwarding devices are programmed by control plane elements through well- defined SI embodiments. The control plane can therefore be seen as the „„„network brain.‟‟‟ All control logic rests in the app applications and controllers, which form the control plane.

3. Management Plane (MP): The management plane is the set of applications that leverage the functions offered by the NI to implement network control and operation logic. This includes applications such as routing, firewalls, load balancers, monitoring, and so forth. Essentially, a management application defines the policies, which are ultimately translated to southbound-specific instructions that program the behavior of the forwarding devices.

b) Firewall

Firewalls are network devices whichenforce the security policy of an organization. A firewall is interposed between two networks to filter traffic between them according to the security policy. A firewall provides security protections by performing rule-based control on packets. Firewalls can be implemented in either hardware or software, or a combination of both. A hardware firewall can be a piece of standalone equipment or a component of a broadband router. Hardware firewalls are typically deployed at the major gateway connecting a protected intra-net and the rest of the network. A software firewall is a software program running on a computer, which protects a computer by constraining the external attempts to gain access to the computer. The hardware firewalls offer better protection and performance than software firewalls. The functions of firewalls range from stateless packet filters to stateful application gateways. The stateless packet filters apply filtering rules to accept or reject individual packets without examining the relationship among packets.

The architecture defined a firewall integrated in router. The firewall determines which inside services may be accessed from the outside, which outsiders are
permitted access to the permitted inside services, and which outside services may be accessed by insiders.

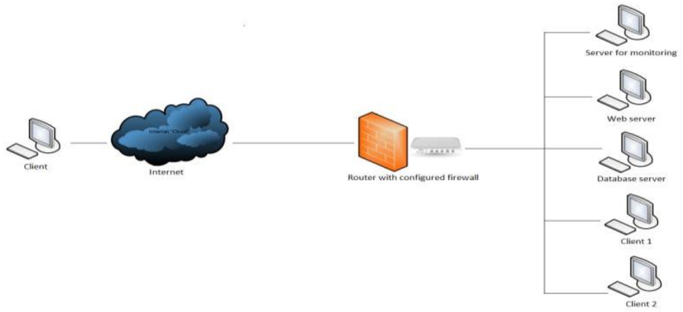


Fig: 2. Architecture of Firewall

III. PROBLEM DEFINITION

1) Unauthorised Access

This issue relates to access control. The main characteristic of SDN is centralized control. When we introduced Distributed SDN Controller, there is possible for multiple controllers to access the data plane. Similarly applications from multiple sources (3rd party apps) may link to the multiple controllers. The controller will give the access to applications so that the applications can read/write the network. If an attacker impersonated a controller/application, it could gain access to network resources and can change the network operation.

2) Distributed Denial of Service (DDoS)

DDoS attack is an effort of an attacker to makea machine or resource unavailable to the intended users. DDoS attacks are sent by two or more persons, or bots, In SDN DDoS can happen at Control Layer as well as Infrastructure Layer. The attacker can flood the controller with packets and flood the infrastructure layer with flow table where limited resources only used there.

3) Handling Traffic

The SDN controller is responsible for path selection and therefore all policy information resides at the controller. The traffic engineering problem that we consider is motivated by scenarios where SDNs are incrementally deployed in a network. In such a network, not all the traffic is controlled by a single SDN controller.

4) Anomaly in OpenFlow Table

OpenFlow provides a standardized way of managing traffic in switches and of exchanging information between the switches and the controller. The OpenFlow switch is composed of two components. Firstly, it contains one or more flow tables responsible for maintaining the information required by the switch in order to forward packets. Secondly, It is an OpenFlow client, which is essentially a simple API allowing the communication of the switch with the controller. The OpenFlow-Enabled switch, contains one or more flow tables and securely communicates with a controller via OpenFlow protocol. Flow tables consist of flow entries, each of which determines how packets belonging to a flow will be processed and forwarded. If the flow table entries get conflicted then

5) Conflict Rules in Firewall

A rule conflict refers to rules that contradict thesecurity policy. Particularly, if a set of rules contravene previous rules it says a conflict or policy violation. In general words, a firewall policy has a blacklist and a whitelist to enforce an achievement. There are four kinds of rule conflicts after misconfiguration.

Shadowing: a rule r_y is shadowed if there is a previous rule r_x that matches the same header but have different action.

Correlation: two rules r_x and r_y are correlated if some headers that match r_x also match r_y, but those rules have different actions. e.g. Rules 1 and 3 are correlated.

Redundancy: two rules are redundant if both perform the same action over the same packet header. e.g. rules 6 and 7 are redundant.

Generalization: a rule that matches r_x is a particular case of another matching r_y, but they perform different actions. e.g. rule 2 is a generalization of rule 1.

Current SDN controllers, such as Floodlight, offer a framework to develop, test and run applications that control the network operation, including the firewall function. However, they are not able to validate firewall policies, detect conflicts neither avoids contradictory configurations on network devices. Some compilers only detect conflicts by a subset of the language hence, it cannot detect conflicts related to contradicting rules with security controls.There are four types of conflicts occurring in rules

i) Shadowing ii) Correlation iii) Redundancy
iv) Generalization

IV. Solutions To The Issues

Before find out the solutions to the issues in SDN one should know the following concepts which might be useful to understand the operation of Firewall in SDN.

1. OpenFlow Protocol
2. Firewall Controller with input Firewall
3. Distributed Based Firewall

1. OpenFlow Protocol

OpenFlow consists of a set of protocols and Application Programming Interface (API). The protocols are divided into two parts.

- The OpenFlow protocol, also called the wire protocol. This defines a message structure that enables the controller to add, update, and delete flow entries in the OpenFlow Logical Switch flow tables as well as to collect statistics.
- The OpenFlow management and configuration protocol that defines an OpenFlow- enabled switch as an abstraction layer called an OpenFlow Logical Switch. This enables high availability by allocating physical switch ports to a particular controller

An OpenFlow-enabled switch consists of a group table and one or more flow tables, one or more OpenFlow secure channels that connect the switch to an external controller, and an OpenFlow protocol that defines the control messages between the switch and the controller.

In this paper, we describe a SDN-oriented prototype of a distributed stateful firewall. The prototype is implemented based on OpenFlow which is a communication protocol used to implement SDN designs on networking devices. This SDN-oriented prototype consists of an OpenFlow- enabled inexpensive “dumb” switch and a firewall controller. The relatively inexpensive “dumb” switch forwards traffic based on the control decision made by the controller, and the firewall controller runs the firewall software.

2. Firewall Controller

The basic structure of a SDN-oriented stateful firewall includes an OpenFlow-enabled switch and a firewall controller. An OpenFlow-enabled switch runs at the gateway connecting a network under protection and the rest of the network. The firewall controller can be potentially hosted anywhere in the network. The security rules are specified in the flow table which are maintained in both the OpenFlow- enabled switch and the firewall controller

3.Distributed Based Firewall

Dkstributed Based Firewall includes an OpenFlow-enabled switch and a distributed firewall controller. An OpenFlow-enabled switch runs at the gateway connecting a network under protection and the rest of the network. The firewall controller can be potentially hosted anywhere in the network. The security rules are specified in the flow table which are maintained in both the OpenFlow-enabled switch and the firewall controller.

Solitions

The Solution to the issues in SDN is implementing a Distributed based Stateful firewall without affecting Openflow Protocol where a firewall Controller is also Present in the SDN control layer which also concentrate more on input flow firewall to find the Conflicts in firewall rules A different approach in which they form a distributed firewall on every forwarding devices in the network. The controller will install all firewall rules in the flow tables of every switch. Obviously, this method is more complicated in configuring the firewall because installing rule commands are exchanged between controller and data plane. On the other hand, the workload for traffic filtering is totally migrated off the controller, and the unified firewall rules in global scale make the whole network less sensitive to topology changes.

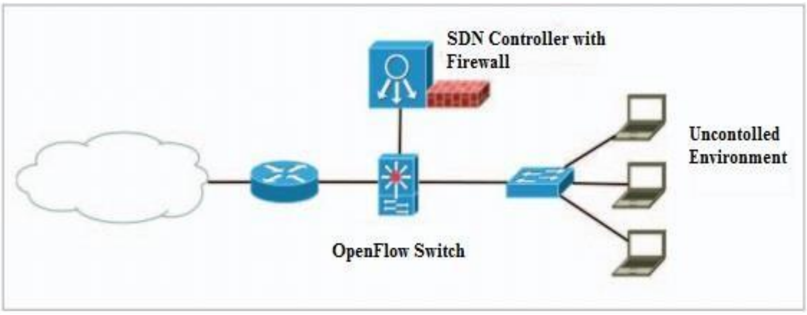
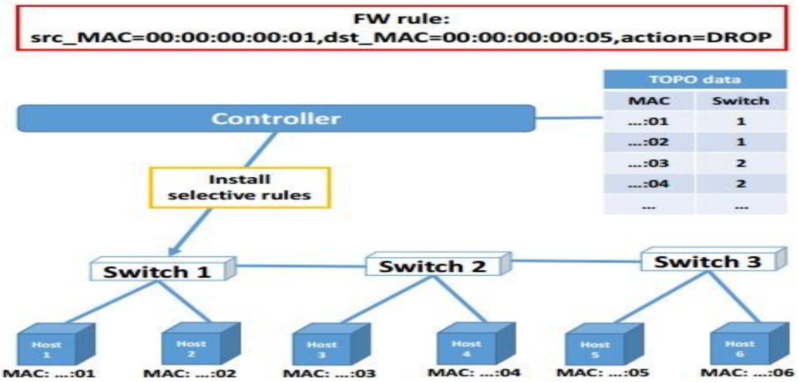


Fig: 3. SDN Enabled Firewall


FW rule:
src_MAC=00:00:00:00:00:01,dst_MAC=00:00:00:00:00:05,action=DROP

TOPO data	
MAC	Switch
....:01	1
....:02	1
....:03	2
....:04	2
...	...

Fig: 4. Distributed Firewall installing selective rules in Switches.

The basic structure of a SDN-oriented stateful firewall includes an OpenFlow-enabled switch and a distributed firewall controller. An OpenFlow-enabled switch runs at the gateway connecting a network under protection and the rest of the network. The firewall controller can be potentially hosted anywhere in the network. The security rules are specified in the flow table which is maintained in

both the OpenFlow-enabled switch and the firewall controller. An entry in the flow table specifies the security rule for handing a traffic flow. The switch acts as a simple packet-pusher based on the security rules defined in its flow table. The firewall controller makes use of its flow table to keep track of the control decisions on traffic flows. A designated communication channel is maintained between the switch and the firewall controller. Through this channel, the switch sends the information about the unidentified traffic flows to the controller for inspection, and the controller sends the control decisions to the switch. When the switch cannot match a packet to a rule in its flow table, it sends the packet to the controller for inspection. After inspection, the controller informs the switch about its control action on a flow, and it also memorizes the control decisions in its flow table. The switch can also update the controller about the statistics. The general operation of a distributed firewall is as follow. First, on the setup phase of the firewall, the controller will send OpenFlow commands to install firewall rules in network switches. We consider the firewall on network layer 2, so each firewall rule is a flow entry consists of source MAC address matching field, destination MAC address matching field, and the action set to DROP action. After setup phase, every packets go through the switches will be matched against flow table entries (including the firewall rules). If the packet matches one of the firewall rule (same source and destination MAC addresses), the switch will drop the packet immediately. When the switch cannot match a packet to a rule in its flow table, it sends the packet to the controller for inspection. After inspection, the controller informs the switch about its control action on a flow, and it also memorizes the control decisions in its flow table. The switch can also update the controller about the statistics. The ability of class- based security control has been implemented on the firewall controller. When multiple traffic flows can be categorized into a class, the default control action on the class of flows can be specified in one entry in the flow table of the controller. Two modes have been implemented to specify the actions on the classes: the permissive mode and the restrictive mode. The permissive mode means selective denial of flows, and the restrictive mode means selective permission of flows. The two modes are specified as allow and denial rules in the flow table used by the controller. An allow rule can also associate with a maximum number of flows that can be allowed.

A firewall controller is also able to limit the incoming traffic on a firewall and allow the servers to deal with the already established connection. This will let the users finish their work and the new users will be able to connect to the server and it also adopts a “first deny last allow” methodology to determine a control action on a flow. A controller prioritizes matching a flow to a denial rule over an allow rule. The precedence of the deny rules over the allow rules alleviates the security risk resulted from possibly conflicting rules in the flow table of the controller. After the firewall controller has determined a control action on a flow, it installs a high-priority rule entry in its flow table to reflect the control decision. The rule entry is subsequently sent to the switch. A rule entry is also associated with an expiration time duration during which the rule remains in effect.Upon expiration, a rule entry is removed from the flow table. Expiration allows the firewall controller to reclaim the allow rules that are associated with specified limitations. The functionalities of the firewall controller are included in four major modules: core, rule processing, configuration, and console. A firewall controller has to initiate and run an instance of the four modules for every switch that is under its coverage. The core module acts the main control unit with the assistance provided by other modules. A flow table is maintained by the core module for regulating the traffic flows running through a covered switch. The rule processing module translates the rule specification to a collection of rule data structures, matches traffic flows to rules, and maintains the structure of the flow table. The configuration module translates the user- specified configuration into a data structure that is recognizable by the core module. By this flow table, traffic will reduced and input flow will reduce the DDoS attack

Therefore, every hardware devices in the network will behave as firewall and filter out unwanted traffic. It can be said that the flow concept is the backbone of OpenFlow and SDN, thus we decide to go with the distributed firewall solution.

V. Implementation

Firewall architecture comprises a module to parse structure into Alloy language, a module that abstracts the topology, a module that handles firewall rules, a module that creates trees of data-fields, and I/O interfaces: RESTlet and interpreter. Firewall is an independent application and can run remotely to the controller. It serves and consumes information from/to the controller. Firewall needs to run in parallel with Alloy version 4 and UNSAT solver. Firewall, at first instance, has a RESTlet that gets information from the Floodlight controller. This information includes topology nodes and links, and the set of rules in the firewall application. Then, Firewall resolves nodes, links, and firewall rules, and builds its own representation on JAVA data structures. Network topology is also represented on its structure and creates forwarding tables for each device with information available through the Floodlight controller. Firewall rules are inspected. Fields from all rules are analyzed, Firewall identifies relations over fields and builds data-trees.This procedure maps variables and rule definitions into sets to reduce the amount of variables. After, Firewall uses its parser module to write rules, topology, paths, and flows into sets and relational logic expressions. the model. Additionally, Firewall creates the functions to classify filter-field relations, and rule conflicts. The id of each rule is represented as a sequence of elements used to illustrate priority. This output is recorded into a .als file which is also possible to run from the Alloy interface. Firewall uses the Alloy-Java-API interface to process the model, and receives the solver output. An unsatisfiable result is an abstraction of a conflict. The solver, in this case unSATcore, executes verification clauses, denoted by check-assert into Alloy model. Finally, results from the SAT solver are interpreted by Firewall to identify counter examples found by the solver. The interpreter decodes the output and identifies the set of fields, rules, and nodes that are received in the counter example and identifies the conflict. Finally, rules in conflict are shown to the console in flood light controller.

VII.CONCLUSION

In this paper, we present a SDN Oriented Disrtibuted firewall, towards efficient filtering rule setup and effective redundancy traffic prevention. The security rules are specified in the flow tables in both the OpenFlow-enabled switch and the firewall controller. The firewall controller is in charge of making control decisions on unidentified traffic flows. The control decisions are specified as control rules in the flow tables. The switch enforces the control decisions by regulating the traffic flows based on the control actions specified in its flow table. The method was implemented and in firewall. It does not cause an increase of the firewall load but prevents the server from overload by keeping the server"s load at a stable level during the attack. The proposed method may be successfully implemented on any firewall- type devices. The Alloy read policy configuration directly from SDN Floodlight controller , and builds abstract representation of policies. Our tool exploits relational logic to explore configuration conflicts in firewalls The translation made by the parser is fundamental for the performance of the SAT-solver and general behavior. We use Alloy to create better translation of CNF Only the mechanism of regulating the number of packets in a single time slot mayrequire some adjusting. The mechanism should be selected for a particular server, depending on its capability and this can be a subject of further research.

REFERENCES

[1]. Diego Kreutz, Paulo Esteves Verrissimo and
[2]. Siamak Azodolmolky, “Software-Defined Networking:A Comprehensive Survey”, Proceedings of the IEEE | Vol. 103, No. 1, January 2015.
[3]. Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer, “A Survey of Security in Software Defined Networks”, IEEE COMMUNICATION SURVEYS & TUTORIALS.
[4]. Ferney A. Maldonado-Lopez∗†, Eusebi Calle†
[5]. and Yezid Donoso, "Detection and Prevention of
[6]. Firewall-Rule Conflicts on Software-Defined Networking", IEEE Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop.
[7]. Qiao Yan, F. Richard Yu, Senior Member, IEEE, Qingxiang Gong, and Jianqiang Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, IEEE Communications Surveys & Tutorials
[8]. Alaauddin Shieha University of Colorado Boulder, "Application Layer Firewall Using OpenFlow".
[9]. Thuy Vinh Tran, Heejune Ahn , "A Network Topology-aware Selectively Distributed Firewall Control in SDN", Information and Communication Technology Convergence (ICTC), 2015 International Conference on Year: 2015
[10]. Jake Collings , Jun Liu , “An OpenFlow-based Prototype of SDN-Oriented Stateful Hardware Firewalls”, 2014 IEEE 22nd International Conference on Network Protocols
[11]. Lukasz Apiecionek , Wojciech Makowski , “Firewall rule with token bucket as a DDoS protection tool” IEEE 13th International Scientific Conference on Informatics · informatics"2015.