2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015)
Dehradun, India, 4-5 September 2015

Intrusion Detection System based on Software Defined Network Firewall

Mohd Abuzar Sayeed*, Mohd Asim Sayeed† and Sharad Saxena‡
* Research Scholar,CSED, Thapar University, Patiala, Punjab, India., abuzar.sayeed@gmail.com
†R and D Engineer, Robonest Enterprises, Lucknow,Uttar Pradesh, India, mohdasimsayeed@gmail.com
‡ Assistant Professor, CSED, Thapar University, Patiala,Punjab,India, sharad.saxena@thapar.edu

***Abstract*—Software Defined Network is an architecture that focuses on the separation of control plane and data plane in order to make networks programmable and scalable. Currently Openflow is the most widely used SDN protocol. It has provided flexibility to the networking environment and had made it simpler and easy to optimize. SDN is a major area of research however; in the current scenario the field of security is relatively under exploited. The paper describes an intrusion detection mechanism for Openflow based Software defined networks. The study focuses on developing a packet filtering firewall over a Software Defined Network controller namely Floodlight and the application of association rules to find the patterns among the data passing through the firewall. The patterns recorded serve as the motivation behind the development of an Anomaly based intrusion detection mechanism.**

Keywords: SDN, Openflow, Firewall, IDS, Floodlight.

I. INTRODUCTION

Before the advent of Software Defined Networking the network industry was vertically integrated i.e. specialized hardware and over that a specialized control plane and specialized features. This vendor specific hierarchy which comes with each logic hard coded into the chips and leaves user with very less options, configurations and customization [1]. Scalability is a major disadvantage of the current trend and the expansion of the service cards to support more flows or service states is almost impossible. The forwarding mechanism needs an upgrade of processing. A majority of network equipment today cannot evolve because they need a hardware upgrade for it. The current network equipment is complex and this complexity makes it fragile and costly [2].

Software Defined Network provides a solution to the problems of current network technology. The abstraction which SDN provides overcomes the issues of flexibility, scalability and programmability. SDN separates the control plane and the data plane. The control plane is where all the logic is applied and the forwarding plane or the data plane sits dumb and forwards traffic according to its flow table. This separation and necessary abstraction provided by the SDN makes it possible to program a network by simply writing control programs without worrying about the underlying hardware [3].

The key component and todays standard for SDN is the Openflow protocol. Openflow provides with the distinction between the control program and the underlying data or forwarding plane. It also provides with a secure channel between the Openflow switch and the controller. Openflow makes the network programmability possible [4]. A controller can be described as a software system that helps in centralized management and control of the ephemeral network or configuration state. More precisely a controller can be considered as a high level model that represents and controls the network policies. Most of the controllers are similar to the NETCONF or RADIUS models [5]. Since a controller is must between forwarding hardware and the control programs we will use Floodlight a java based controller for the experiment [6].

II. BACKGROUND STUDY

A. OpenFlow

The slow innovation in the field of networking is because of the hardware and protocols that make it impossible to experiment in real time. With the present networking scenario its almost impossible to implement and test new protocols and application programs. There is a network that makes network programmable by allocating a part of the network to the researchers. With GENI a researcher can program its part of the network that can include hardware and software components [7]. The NetFPGA, a PCI card is advancement in the field of research. The card is programmable and can process four Gigabit Ethernet ports. But the major con of this approach is it can process traffic from only four ports [8]. The driving force behind the innovation of Openflow was to develop a protocol that separates the real time traffic from that of the experimental traffic. A protocol that runs at line rate and over which other interesting features can be developed as application programs. An Openflow switch constitutes three main elements, a table of flows i.e. a flow table, a secure tunnel via which controller communicates with the switch and the Openflow Protocol that is the de facto standard for communication between controller and switch [9]. An Openflow switch is like a dumb data forwarding element and behaves as dictated by the control program. Figure 1 explains the working of an Openflow switch. Hybrid switches or Openflow enabled switches are also are also available.

B. Anomaly Detection

Intrusion Detection System can be described as a control program that identifies and logs the threat or unexpected behavior of the network traffic. Anomaly detection can be summarized as a comparison between the average recorded

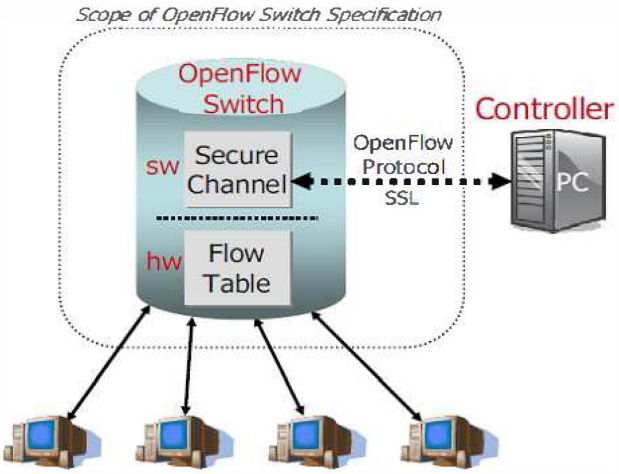

Fig. 1. An Openflow Switch [8].

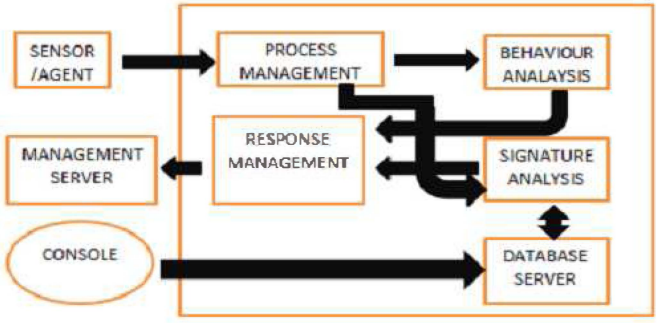

Fig. 2. Taxonomy of an intrusion detection system [11].

behavior and the current observed behavior. The data log is the key component of the IDS and is used for the comparison and validation of the unexpected behavior. Data log can be generated by simply monitoring a host or a network device [10]. A hybrid technology can also be employed which monitors both host and device simultaneously. Network Based Analysis (NBA) can also be used for DDoS attacks or traffic pattern analysis. Anomaly in itself means something abnormal. Anomaly Based IDS works by generating a profile first. This profile generation may take time, sometimes even weeks. Network is meant to change drastically with time so dynamic profiles are maintained mostly as static ones get out-dated with time. Anomaly detection can employ a Statistical Based Method to record user profile with time or a Distance Based Method when data is difficult to collect in distributed environments [10][11]. Rule Based and Model Based are similar approaches in which we have defined the normal behavior of the system or created a profile of the normal behavior of the system which is later used for comparison and intrusion detection purpose [12].

Figure 2 describes an Intrusion Detection System. The sensor has the sole job of monitoring everything. The management server receives information from everywhere and controls the sensors. Console is the user interface between IDS and the system administrator. Database server serves as repository.

TABLE I. OPENFLOW FLOW TABLE

In PORT	VLAN ID	Src MAC	Dst MAC	Dlink Type	Source IP	Destination IP	Network Protocol	Src-Port	Dst-Port	Action

C. Apriori Algorithm

The hidden patterns that occur frequently and generating association rules from those patterns have always been a classical field of study. The patterns may include item sets, structures, records etc. In current environment a lot of prediction or suggestions mechanisms are based on data mining. The Apriori algorithm is one of the classical algorithms that mine transaction databases for finding Boolean association rules. The basic ideology behind this algorithm is that a subset of a frequent occurring set of items must also be a frequent set of items. The infrequently occurring items are pruned using the algorithm.

The following pseudo code describes the basic apriory algorithm [13].

1) Transaction database is scanned and item sets are generated.
2) The Y item sets generated in previous step are compared with minimum support count and sets of frequent items are generated.
3) The candidate Y+1 item sets are generated from the frequent Y item sets generated in previous step and hence frequent K+1 item sets are generated.
4) Pruning is applied on candidate Y+1 sets of items to generate frequent item sets of Y+1 item. Pruning deletes all those item sets whose support count is less than minimum support count. The steps are repeated until no new candidate set can be generated.

For mining association rules from frequent item set, we create all the subsets of the frequent item sets. Make sure all non-empty subsets are created. For each non-empty subset S of frequent item set I.
$S \rightarrow (IS)$
If and only if support count (I)/support count(S) ¿= minimum confidence threshold [13].

III. IMPLEMENTATION

The design of Openflow based intrusion detection system begins SDN Firewall implementation. It is necessary to have an access control mechanism that filters and logs the data. The firewall consists of a learning switch and a floodlight firewall controller. Flow table is modified and rules of the firewall are installed into it. Table 1 describes a typical Openflow flow table. Mininet is used to generate a virtual network for the test environment. Mininet runs a kernel and an OpenFlow switch to create a real time virtual environment on a single computer, virtual machine or a cloud [14]. Figure 3 and Figure 4 show the topology used in the simulation and its creation.

A. Firewall Features

The firewall consists of six features ADD, DELETE, MODIFY, SHOW, SETPRIORITY, and TIMEOUT. The keywords are mandatory for defining, editing or deleting a rule.

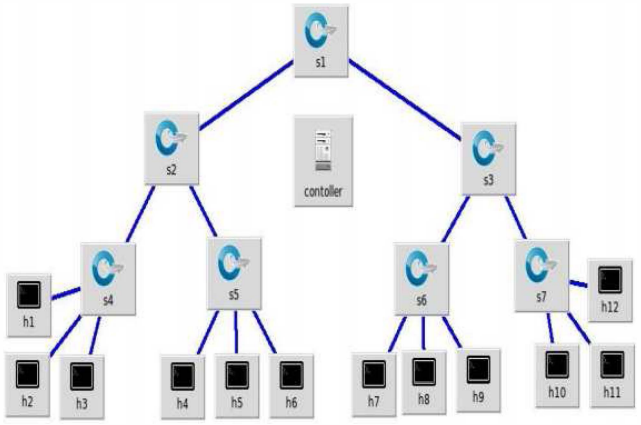

Fig. 3. Simulation topology.

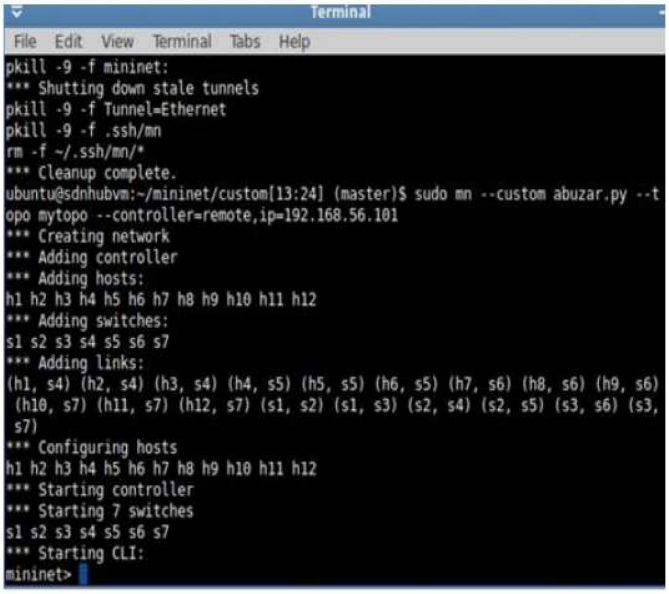

Fig. 4. Mininet Console.

These keywords combined with other attributes define a SDN Firewall. ADD command takes in the attributes as parameters and creates a new rule accordingly. Priority is optional and if left undefined minimum priority is considered. The rule can be given an name that is optional. DELETE command takes name of the rule as input and deletes the above rule. MODIFY takes name of the rule as input and modifies the rule. SHOW command if applied with the rule name shows that rule else shows all the rules by default. SETPRIORITY command takes the name of the rule as input and resets or sets the priority of that rule. TIMEOUT command sets the time after which the rule expires. Timeout time is measured in milliseconds.

B. *Firewall Attributes*

- Nw-proto takes the name of the protocol as input. The name that is a string value is not case sensitive.
- Src-ip/Dst-ip takes an IPv4 address/mask as input.
- Port takes the input port as input. -MAC/Dst-MAC takes source and destination MAC addresses as input.

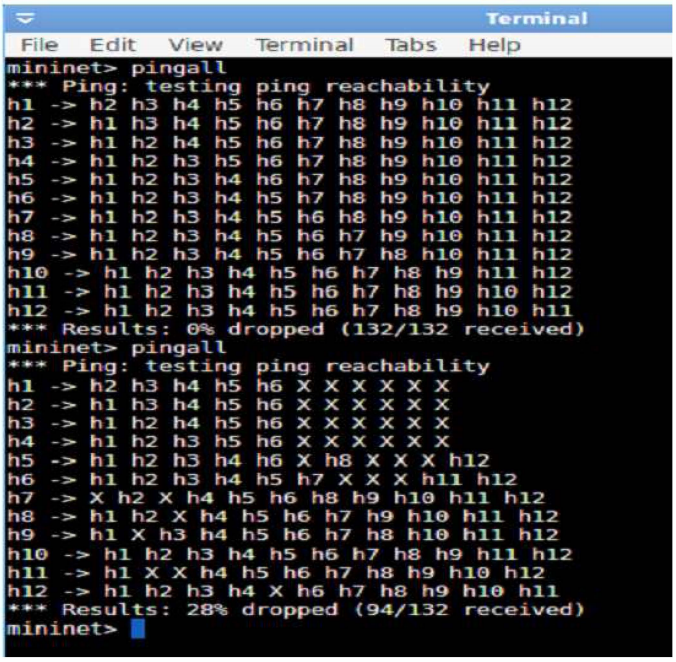

Fig. 5. Application of rules over ICMP packets.

- Vlan-id takes vlan-id as input.
- Vlan-pri is used for Vlan priority.
- Tp-src/Tp-dst takes TCP/UDP port numbers as input. -type describes the type of Ethernet we are using.
- Switch-id is the id of the switch.

C. *Firewall Rules*

The following sample rules describe how to write a firewall rule for our controller. Figure 5 shows the application of rules over ICMP packets. The rule is same as the fifth rule. Second rule allows all the ip traffic between source and destination ip address. Third rule allows all the traffic to the mentioned switch id.The fourth rule is stricter as it dictates the mac and ip addresses of both the communicating hosts.

- RULENAME ,"src-ip": "192.168.2.1/24", "dst-ip": "191.168.2.2/324", "dl-type":"ARP", action: deny.
- RULENAME, "src-ip": "191.168.2.1/24", "dst-ip": "191.168.2.2/24", "nw-proto": "ICMP",action: allow.
- RULENAME, "switchid": "12:34:56:12:34:56:01:01", action : allow.
- RULENAME, "src-ip": "191.168.2.2/24", "dst-ip" : "192.168.2.1/24","src-mac": "77:81:93:67:18:1a", "dstmac" : "12:34:56:12:34:56:01:0a, action: allow.
- RULENAME,dl-type:ARP,action: deny.

IV. PROPOSED IDS SYSTEM

Initially we define the dataset that consists of expert suggested or previous data and data from the audit of our own experimental setup i.e. our access control mechanism. A profile of normal and abnormal signatures is built over

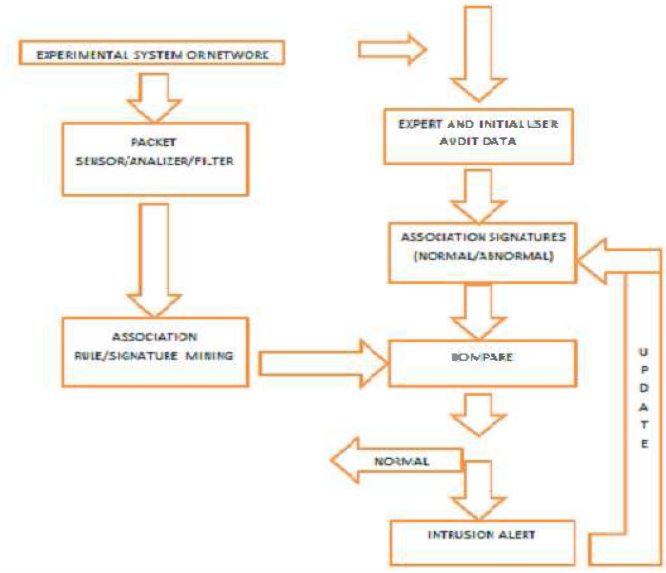

Fig. 6. Architecture of Intrusion Detection System.

this data. We use Apriori algorithm for signature mining or association rule mining. The real time firewall filtered data is mined and compared with the predefined rules. The deviations from the normal behavior are considered potential threats and a potential threat alert is raised. The IDS system is updated accordingly. Figure 6 describes the proposed IDS system,

The following algorithm is used for signature and association rule mining.

ALGORITHM: (Mining association rules by Apriori algorithm)
INPUT: Expert/Initial Database (EID), Minimum Support Count (α), Minimum Confidence Threshold (β).
OUTPUT: Set of association rules/Signatures.
MEATHOD:

1) Search the complete EID and select the items whose support count $(\mu) \geq \alpha$
2) Determine the Y+1 sets of frequent and candidate items from the item sets of Y frequent items. Pruning is done for determining the frequent item sets by deleting those whose $(\mu)\alpha$. Repeat until candidate list is exhausted.
3) Create non empty subset S of frequent item set I, Then $S \rightarrow (I - S)$, Only if $\mu(I)/\mu(S) \geq \beta$.

V. CONCLUSION

Security has always been one of the major concerns of the network industry. Traditional networks had specialized hardware, over the hardware specialized protocols and applications that made the industry very rigid in terms of flexibility and evolution. With the advent of software defined networks the proprietary industry became programmable and flexible. This flexibility on one hand becomes an attractive target for the attackers but on the other hand if well managed and properly designed this environment becomes more resilient to attacks and other vulnerability. The SDN in itself relies only on the secure channel for security or replication. But programmability also provides the users with the independence of designing their own security mechanisms and application. The IDS system discussed in the paper is one such approach to develop a self-improving security mechanism for the Openflow environment. The proposed system when tested in real time can produce significant results and provide an insight to the development of new security mechanisms and identifying new threats and vulnerabilities. References

REFERENCES

[1] N. Feamster, K. Hyojoon, Improving network management with software defined networking, Communications Magazine, IEEE, vol.51, no.2, pp. 144- 119, February 2013.

[2] T. D. Nadeau and K. Grey, Centralized and distributed control and data planes, in SDN: Software Defined Networks, Ist Ed, Sebastopol: OReilly Media,Inc,2013, pp. 9-44..

[3] Software-defined Networking (SDN) Definition [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition.

[4] OpenFlow [Online]. Available: https://www.opennetworking.org/sdn-resources/openflow.

[5] T. D. Nadeau and K. Grey, SDN Controllers,in SDN: Software Defined Networks, Ist Ed, Sebastopol: OReilly Media, Inc,2013, pp. 71-93.

[6] Floodlight [Online]. Available http:// www.projectfloodlight.Org / floodlight.

[7] Global Environment for Network Innovations [Online]. Available: http://www.geni.net/?page_id=2.

[8] NetFPGA: Programmable Networking Hardware [Online]. Available: http://netfpga.org/site.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L.Peterson, J. Rexford, S. Shenker, and J. Turner, OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 6974, 2008.

[10] White paper, Intrusion Detection: A Survey, ch.2, DAAD19-01, NSF, 2002.

[11] K. Scarfone, P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), NIST Special Publication 800-94, Feb. 2007.

[12] F. Sabahi, A. Movaghar, Intrusion Detection: A Survey, in Third International Conference on Systems and Networks Communications, 2008, pp.23-26.

[13] Association Rules [Online]. Available: http:// saedsayad.com / association_rules.htm.

[14] Mininet [Online]. Available http://mininet.org.