ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

Enhanced Malicious Switch and DDoS Attack Detection in SDN

Akshaya Arunan, GEC, Trivandrum[1] and Simi Krishna KR(Assistant Professor), GEC, Trivandrum.[2]

Abstract— Software defined network is a centralized system where the system allows the network administrators to problematically initialize, control, change and manage the network automatically.Here the network administrators code into the controller which like the brain of SDN. The switches are just forwarding devices. But if any switch is captured by an adversary anytime, it may mislead the controller to make inaccurate decisions risking the network. In SDN as the control plane is separated from the data plane, it allows the network to proper better than the traditional networks. But due to this split, there is a high risk of potential DDoS attacks.Therefore, this paper elaborates on these problems and uses various methods like secured threshold method and Max SPRT to detect the malicious OpenFlow switches and protection against DDoS attacks.

I. INTRODUCTION

The traditional networks are complex and hard to manage. The control plane and the data plane are in routers. And each router makes independent decisions based on its routing table and maintains them autonomously, or it maybe manually set by administrators. Network operators need to configure each individual network devices separately using low-level and often vendor-specific commands. Also the control plane and the data plane are bundled inside the networking devices reducing their flexibility and hindering innovation and evolution of networking infrastructure.

Thus, SDN is proposed to replace traditional networks. SDN is an emerging networking paradigm that gives us hope to change the limitations of the current network infrastructures.

SDN is a more centralized approach, in which the controller communicates with switches via OpenFlow protocol[1]. The controller is a high powered server, an importance piece of software. It is just like a boss role in SDN, and each switch works under the instructions provided by it. Switches only have the storing and forwarding functions. However, the controller knows all the information of the networks, including topological structure, the bandwidth between neighboring switches and so on [1].

Since SDN does not need any special hardware it reduces the network costs. SDN provides to write a program to program the network. The control plane does not need to know as much as network operating system. Thus virtualization layer helps in simplifying for the coders by hiding.

If one switch is becomes malicious, it will start sending inaccurate messages to the controller. For instance, a malicious switch may inform the controller that its processing power and its bandwidth are very huge, which may mislead the controller to assign more flows to the switch[1]. This leads to a large amount of packet loss, which will have a terrible effect on the SDNs performance.

Here are some main characteristics of SDN:

1) Directly programmable
2) Agile
3) Centrally managed
4) Problematically configures
5) Experimenting and research is not expensive
6) Fast upgrades

II. RELATED WORKS

A. Software Defined Networking: A comprehensive Survey[2]

Software-defined networking (SDN) [3], [4] 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 networks control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane)[5]. Second, with the separation of the control and data planes, network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller (or network operating system1), simplifying policy enforcement and network (re)configuration and evolution [5].

A simplified view of this architecture is shown in Fig. 1.

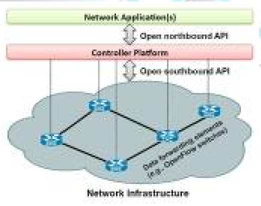


Fig. 1. SDN Architecture[2]

[9] discussed about a method, In vehicular ad hoc networks (VANETs), because of the nonexistence of end-to-end connections, it is essential that nodes take advantage of connection opportunities to forward messages to make end-to-end messaging possible. Thus, it is crucial to make sure that nodes have incentives to forward messages for others, despite the fact that the routing protocols in VANETs are different from traditional end-to-end routing protocols. In this paper, stimulation of message forwarding in VANETs is concerned. SDN as a network architecture has four pillars:

1) The control planes and the data planes are decoupled and the functionality of control plane is removed from

[1] Mrs Simi Krishna KR is with Faculty of Network Engineering, Information Technology, Government Engineering College, Barton Hill, Trivandrum simi.krishna

network devices that later becomes simple (packet) forwarding elements.

2) Forwarding decisions are flow based and not destination based. A flow is a set of packet field values acting as a match (filter) and a set of actions (instructions).
3) The control logic is moved to an external entity called as the SDN controller or NOS.
4) The network is programmable through various software applications running on top of the NOS which interacts with the underlying data plane devices in SDN.

B. A Survey on SDN Programming Languages: Towards a Taxonomy[6]

1) SDN Programming: To understand the SDN programming, it is important to recall its basic operation [7]: When a network device receives a new packet, it searches in its flow tables to find an entry matching the packet header fields (e.g. destination MAC or IP address). If no match is found, the device forwards the packet to the controller.

The applications were programmed in the three different levels:

1) Low-level Programming: Low-level Programming is labeled when the networks are programmed directly through CDPI. OpenFlow stands out as the major initiative in CDPI, but it does not make the programming task easier.
2) API-based Programming: Applications implemented in this programming level use the APIs exposed by the controllers.
3) Domain-Specific Language Programming: A DSL is a programming language that is usually restricted to, a particular problem domain rather than offering appropriate notations and abstractions, expressive power.

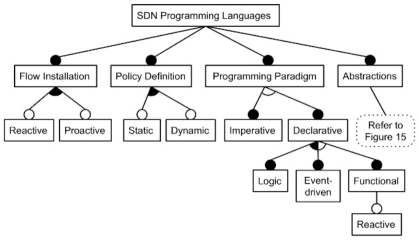


Fig. 2. Top-level classification feature diagram[6]

2) SDN Language Classification: These are:

1) Threshold Value Control
2) User Feedback System

3) SDN Programming languages:

1) Flow-based Management Language: FML is the first SDN programming language. It relies on non-recursive Data log with negation, where each statement represents a simple if then relationship. In a technical report, the language was initially defined as Flow-based Security Language (FSL), however, in the following publication, it was renamed to FML.
2) Nettle: Nettle was the second SDN programming language. Its programming paradigm is based on the principles of Functional Reactive Programming (FRP).
3) Frenetic: The name Frenetic was originally used as the name of a programming language. Over time, it has been used as the name of a project to represent a set of SDN programming languages. Lately the name Frenetic was used again to specify the name of a new language.
4) Procera: As well as Nettle and Frenetic, Procera is based on FRP, using ideas from Yam pa. Despite no information was confirmed that Procera might be an extension of Nettle, given that they were proposed by the same author with similar structural characteristics. Procera was used in various prototype deployments in campus and home networks.
5) Flog: Flog combines ideas from FML and Frenetic. It uses the same idea of FML by proposing a logic programming language to control SDN. From Frenetic, it adopts the idea of dividing the language in three components: a mechanism for querying network state, a mechanism for processing data gleaned from queries (or other sources), and a component for generating rules to be installed on the network switches.

C. Open Sec: Policy-Based Security Using Software-Defined Networking [8]

OpenSec is an Open Flow-based network security framework that allows campus operators to implement security policies across the network. OpenSec provides an abstraction of the network. The operators focus on specifying simple and human-readable security policies simplifying the overall process. Open Sec consists of a software layer on top of the network controller, as well as multiple external devices that perform various security services and finally report the results to the controller. Opensec allows describe the security policies for specific flows which includes a description of the flow, a list of security services that apply to the flow and how to react in case malicious content is found with the network operators[8].

OpenSec has three design requirements:

1) Policies should be human-readable.
2) Data plane traffic should be processed by the processing units (network devices, middle-boxes or any other hardware that provides security services to the network).
3) The framework should react to security alerts automatically to reduce human intervention when suspicious traffic is detected.

D. Traffic-based Malicious Switch Detection in SDN

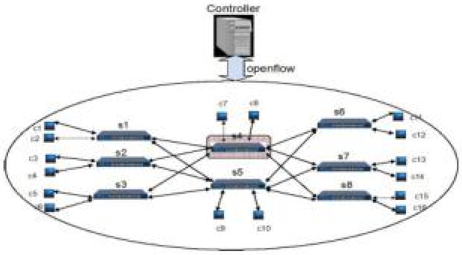


Fig. 3. A Typical SDN Topology[8]

There are 8 switches named from s1 to s8, and 16 hosts.

III. PROBLEM FORMULATION

Here, first state is the DDoS detection problem in the SDN network. Later, the network assumptions and the attacker models is described to evaluate.

Figure evaluates the logical view of the network model. Since a DDoS attack could come from any of the interfaces of any switch, our goal is to detect the attack and locate the potential interfaces that are compromised.[10]

This section elaborates on solutions that can help the controller to detect malicious switches. There are main two steps that we would be implementing to help detect malicious switches.

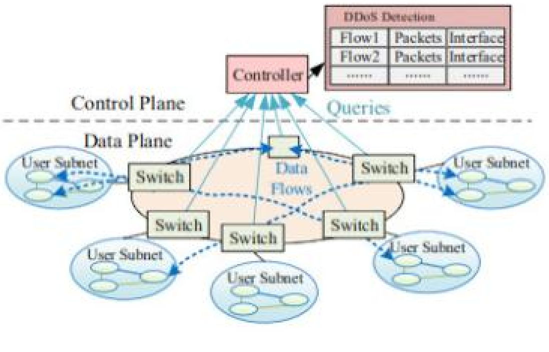


Fig. 4. Network Model[10]

SDN interface switch is defined as a compromised interface if it is connected by the malicious DDoS attackers. Many low traffic flows are likely to be injected into the compromised interfaces by the malicious attackers, with the purpose of triggering a high volume of switch-to-controller messages to overload the SDN controller.

While detection, it is assumed that each switch is capable of obtaining the statistics information of the incoming flows and timely reporting it to the controller. This can be achieved by three methods:

1) Open Flow itself can monitor per-port and per-rule byte and packet counters
2) NetFlow checks the header fields of each packets to see if the packets match an existing flow.
3) sFlow can estimate the byte and packet counts of flows by fetching real-time packet samples from switches.

A. Secured threshold value control

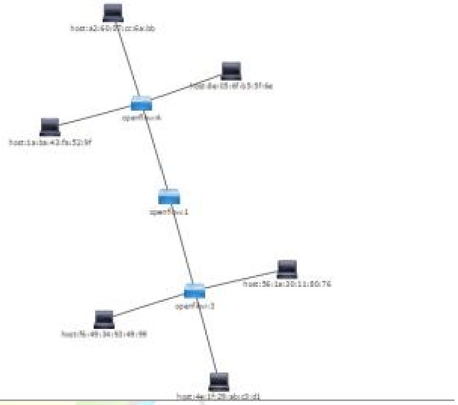


Fig. 5. SDN Topology ODL

There are 3 switches named Open flow 4,1,3, and 6 host. The controller can reach all switches.It communicates with every switch using Open Flow protocol via a secure channel. Suppose s4 is a malicious switch. Assume that the bandwidths of 4-1 is 2Mb/s, the bandwidths of 1-3 is 4Mb/s. If host 1 wants to communicate with host 4, the controller tends to assign the flow across s4-s1-s3. When s4 is captured by an adversary, s4 sends messages to the controller informing that the bandwidths of 4-1 is 10Mb/s. Next, when new packets arrive at s1 from host 1, the controller prefers to forward packets through s4 as it believes that the bandwidth of 4-1 is wider. This leads to a lot of packet loss also creating a terrible influence on the communication quality of the networks. Thus, introducing a threshold value control.

A table is maintained in the controller platform and store maximum traffic-flows of each switch in Open Flow networks.

The Table is maintained by the controller. The MAC fields aims to locate the unique switches in networks. Each switch has a threshold field which the controller finds out.

When the network is initialized, the controller will communicate with switches frequently.The controller knows more than just the maximum traffic-flows threshold values. The information is maintained by the Topology Manager in the controller. The controller is responsible for building the topology of SDN and therefore it must know all bandwidths between every two switches. By checking the table, the controller can find out suspicious switches.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

B. Detection based on Sequential Probability Ratio Test (SPRT)[10]

A flow classification function and an attack detection function is described.

1) Flow Classification: It classifies the data flows in the network. A data flow is considered to be either a normal flow or a low-traffic flow.[10] Upon receiving a statistic report about a data flow, the flow classification makes distinctions.
   After that, the flow classification function reports the results to the attack detection function.
2) Attack Detection based on SPRT[10]: It analyzes the list of observed events to decide whether an interface is compromised or not.
   It can make two types of errors:
   - false positives
   - false negatives

   The SPRT-based detection method can be considered as a one-dimensional random walk.

C. Maximized Sequential Probability Ratio Test(MaxSPRT)[11]

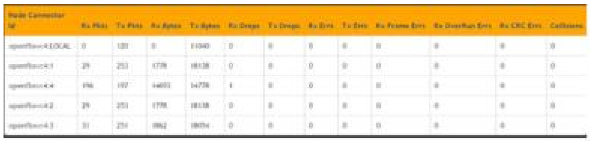
MaxSPRT is a generalized sequential probability ratio test, as defined by Weiss (1953, p. 273). Because we are using a likelihood ratio with a composite alternative, MaxSPRT is also a sequential generalized likelihood ratio test, a term first used by Siegmund and Gregory (1980, p. 1223). MaxSPRT is and extended method of SPRT. It is a method used for continuous surveillance of data. It can be explored for two different probability models using the Poisson and binomial distributions respectively. In general Max SPRT can be used for other distributions such as the hyper geometric, suitable for other types of data. MaxSPRT can calculate critical values.The MaxSPRT was developed in response to direct vaccine safety surveillance needs in the Centers for Disease Control and Prevention (CDC)-sponsored Vaccine Safety Datalink (VSD) and, as such, it is already in practical use [10].

IV. RESULTS AND DISCUSSION

A set of threshold values are alloted to each switches below. This is maintained as a table (Table 1) in controller to match these thresholds to ensure its validity. Here in this table S1 and S4 is alloted a threshold of 3 Mbps and 4 Mbps respectively. Therefor if any switch is captured and the threshold value changes, it would come under the notice of the controller which in turn is then blocked by the controller.

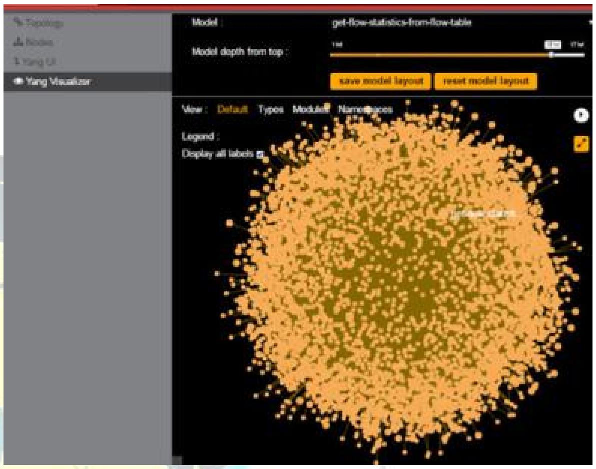
TABLE I
THRESHOLD VALUE OF EACH SWITCH

Yang visualizer showing the statistics of the network flow (fig 7).



Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:4:LOCAL	0	120	0	11040	0	0	0	0	0	0	0	0
openflow:4:1	29	253	1778	18138	0	0	0	0	0	0	0	0
openflow:4:4	196	197	14693	14778	1	0	0	0	0	0	0	0
openflow:4:2	29	253	1778	18138	0	0	0	0	0	0	0	0
openflow:4:3	31	254	1862	18054	0	0	0	0	0	0	0	0

Fig. 6. OpenFlow through Switch S4



Model : get-flow-statistics-from-flow-table

Model depth from top :

Fig. 7. Yang Visualizer for open flow statistics

Table 2 and 3 describes the packet forwarding achieved through switch S3 and S4. Packet is lost through switch S3 from h4 to h3.

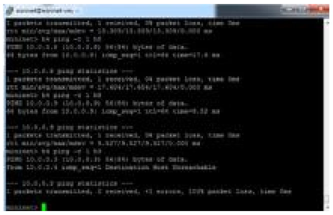
TABLE II
PACKET FORWARDING ACHIEVED THROUGH SWITCH S3 AND S4

Host Name	Src IP Addr	Src MAC Addr	Destination IP Address
h4-h8	10.0.0.4	56:1a:30:11:80:76	10.0.0.8
h4-h7	10.0.0.4	56:1a:30:11:80:76	10.0.0.7
h4-h9	10.0.0.4	56:1a:30:11:80:76	10.0.0.9
h4-h3	10.0.0.4	56:1a:30:11:80:76	10.0.0.3

TABLE III
PACKET FORWARDING ACHIEVED THROUGH SWITCH S3 AND S4

Dstn MAC Addr	Time for pckt frwding
8e:05:6f:b5:5f:6e	17.6ms
1a:ba:43:fa:52:9f	13.3ms
a2:60:07:cc:6a:bb	9.52ms
fe:7d:41:37:ed:e0	Packet lost

Switch Name	IP Address	MAC	Threshold
S1	192.168.56.101	12:c9:cf:f7:f2:4b	3Mbps
S4	192.168.56.104	06:c3:ce:c0:e1:4f	4Mbps



```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = [illegible] ms
mininet> [illegible] ping -c 1 [illegible]
PING 10.0.0.[illegible] (10.0.0.[illegible]) 56(84) bytes of data.
64 bytes from 10.0.0.[illegible]: icmp_seq=1 ttl=64 time=[illegible] ms

--- 10.0.0.[illegible] ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = [illegible] ms
mininet> [illegible] ping -c 1 [illegible]
PING 10.0.0.[illegible] (10.0.0.[illegible]) 56(84) bytes of data.
64 bytes from 10.0.0.[illegible]: icmp_seq=1 ttl=64 time=[illegible] ms

--- 10.0.0.[illegible] ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = [illegible] ms
mininet> [illegible] ping -c 1 [illegible]
PING 10.0.0.[illegible] (10.0.0.[illegible]) 56(84) bytes of data.
From 10.0.0.[illegible] icmp_seq=1 Destination Host Unreachable

--- 10.0.0.[illegible] ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

mininet>
```

Fig. 8. Pinging between hosts.

V. CONCLUSION

In this paper, an efficient detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows is discussed. The method is based on powerful statistical tool, SPRT. Also a secured threshold value controlled is implemented to avoid malicious attacks in SDN.

For future research the first security analysis on the SDN/Open Flow Topology Management Service can be performed. Also ways can be implemented to use Network Topology Poisoning Attacks to exploit the vulnerabilities.

REFERENCES

[1] Xiaodong Du, Ming-Zhong Wang, Xiaoping Zhang and Liehuang Zhu, Traffic-based Malicious Switch Detection in SDN.International Journal of Security and Its Applications Vol.8, No.5 (2014), pp.119-130.

[2] Diego Kreutz, Member IEEE, Fernando M. V. Ramos, Member IEEE, Paulo Esteves Verssimo, Fellow IEEE, Christian Esteve Rothenberg, Member IEEE, Siamak Azodolmolky, Senior Member IEEE, and Steve Uhlig, Member IEEE, Software-Defined Networking: A Comprehensive Survey.

[3] N. Mckeown, How SDN will shape networking, Oct. 2011. [Online].

[4] S. Schenker, The future of networking, the past of protocols, Oct. 2011. [Online]. Available: http://www.youtube.com/watch?v=YHeyuD89n1Y.

[5] H. Kim and N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag., vol. 51, no. 2, pp. 114119, Feb. 2013.

[6] Celio Trois, Marcos D. Del Fabro, Luis C. E. de Bona, A Survey on SDN Programming Languages: Towards a Taxonomy.

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

[8] Adrian Lara and Byrav Ramamurthy, OpenSec: Policy-Based Security Using Software-Dened Networking, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 13, NO. 1, MARCH 2016.

[9] Christo Ananth, Kavya.S., Karthika.K., Lakshmi Priya.G., Mary Varsha Peter, Priya.M., "CGT Method of Message forwarding", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1, Issue 1, August 2015,pp:10-15

[10] Ping Dong, IEEE Member, Xiaojiang Du, Senior IEEE Member, Hongke Zhang, Tong Xu, A detection Method for a Novel DDoS Attack against SDN Controllers by Vast New Low-Traffic Flows, IEEE ICC 2016 Communication and Information Systems Security Symposium,2016.

[11] Martin Kulldorff, Robert L. Davis, Margarette Kolczak, Edwin Lewis4, Tracy Lieu1, and Richard Platt, A Maximized Sequential Probability Ratio Test for Drug and Vaccine Safety Surveillance, May 2015.