International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

SNORT Based Lightweight Intrusion Detection System for SDN

Sanjay N[1], Dr. K. S. Jasmine[2]

P.G Student, Department of MCA, RV College of Engineering, Bangalore, Karnataka, India[1]

Associate Professor, Department of MCA, RV College of Engineering, Bangalore, Karnataka, India[2]

ABSTRACT: There are many new stumbling blocks in the area of Cyber Security. The usage of internet as well as applications, are generating tremendous amount of data exchange between many peers, data centres, and severs. Extending equally intrusions, attacks are in progress. Intrusion detection systems, that exists, are inadequate. Software defined network is basically an architecture which has made the network programmable. By disintegrating control and data planes, SDN has made centralized network controlling possible. Faults make these network very complex. during older version of networks, data plane and control planes were one single device residents, and together for network services. Forwarding plane nothing but data plane is responsible for forwarding packet traffic according to the direction given by the control plane. The direction given by controller is nothing but rules. These procedures/rules are defined by controller/control plane. Hence control plane directs the forwarding plane about the packet flow. SDN is a network architecture with a controller in a remote place. It controls whole network by managing packet forwarding in data plane. Forwarding plane consists of normal packet forwarding devices. These devices pass the traffic packets by referring the flow tables. Flow table is updated by applications running on the top of controller. separation of data plane and control plane have made installation of new services easy. The centralized controlling feature of network has enhanced the level of networking SNORT is a light weight intrusion detection system. NIDS are the important layer of a network. Intrusion creates costly loss. Either as a result of attack or as a commercial NIDS high cost. As a remedy for these problems, SNORT was introduced.

KEYWORDS: Network Intrusion Detection System (NIDS), Software Defines Networks

I. INTRODUCTION

The effectuated system is an Intrusion detection system for organization network, as well as enterprise by utilizing the centralized controlling feature provided by SDN (software defined network). Software defined network is basically an architecture which has made the network programmable. By disintegrating control and data planes, SDN has made centralized network controlling possible. Internet is the one medium which connects computers digitally. It has made any level of task easy. Faults make these network very complex. during older version of networks, data plane and control planes were one single device residents, and together for network services. Forwarding plane nothing but data plane is responsible for forwarding packet traffic according to the direction given by the control plane. The direction given by controller is nothing but rules. These procedures/rules are defined by controller/control plane. Hence control plane directs the forwarding plane about the packet flow. SDN is a network architecture with a controller in a remote place. It controls whole network by managing packet forwarding in data plane. Forwarding plane consists of normal packet forwarding devices. These devices pass the traffic packets by referring the flow tables. Flow table is updated by applications running on the top of controller. separation of data plane and control plane have made installation of new services easy. The centralized controlling feature of network has enhanced the level of networking. SDN optimizes the enterprise network devices. Installed Hardware can be reused to follow the instructions of controller. Low cost hardware can be brought into play with major impact. Managing data of a firm, needs IT department to deploy many VMs (virtual machine), and many applications specifically for processing the services.Implementation of SDN makes managing easy and configurations can be updated with zero effect to the existing network as well as without physical work. Data traffic control is also one of the major issues in networking which is handled and controlled by SDN effectively. The data direction feature, manages the traffic and enhances the effectiveness of the network. Openflow is a framework which enhances the centralization of networking by communicating the control plane and data plane. Openflow is also known as a network protocol. It communicates to devices in network. Virtual networks such as data centers, service providers are the main beneficiaries of software defined networking.

II. RELATED WORK

Virtual networks that are programmable, supports the deployment of service specific programs. Otherwise lack of resources may become a major issue for implementation if it is a conventional network. As mentioned in paper

"Intrusion Detection on Software Defined Networking" by "Tella Nagarjuna Reddy", and "K.AnnapuraniPanaiyappan" [2]. For implementation virtual network devices are used in this system. Virtual network is centralized by deploying OpenDaylight controller. Major intrusion detection system can be classified as anomaly detection or misuse detection. Intrusion detection system is partitioned mainly into anomaly detection and prohibited access to the resources. Paper [3] summarizes the use of SNORT for intrusion detection, and also usage of Open vSwitch. Intrusion detection system will be an additional function for existing network system. which may reduce the quality of service. Paper [4] gives information about effect of SNORT and bro IDS tools on quality of service. The current system uses SNORT for attack detection. A lite weight IDS, cross platform application which will be deployed in under the network where OpenFlow will be the controller. Paper [5] gives a backbone idea of how collaborative IDS can be prepared. Paper [14] has explained interfaces of SDN architecture, which actually helps to understand the architecture, where to deploy application and how SDN works. Important interfaces of SDN are,

- Northbound Interface as shown in Fig 1, itis responsible for data exchange between the SDN and the application running on the top of SDN. Nothing but application driven network. No standards exist for this interface. The frequency, type of data, and whatever they exchange it also doesn't have any pre-defined standard. It purely depended on the type of network, and the network application running. Northbound interface is basically an API which through which the orechestrators and applications can program to handle the networks. Some major features of northbound API are computation of paths, avoidance of loop, security and routing. Openstack quantum is an archestration system which communicates through the northbound API to manage network by programming. Still capability of northbound API is being evaluated.
- Southbound Interface is responsible for enclosing the APIs to the lower layers of the architecture. Southbound API allows controller and switches, routers to communicate. OpenFlow is one of the best-known southbound interfaces. Openflow directs the controller about the interaction between the SDN controller and data plane, so that it can be ready for any changes according to business requirements [15].
- Conventional IP networks and SDN gets interconnected by eastbound interface. One particular controller can handle a specific number of switches under it. But the large-scale network and exponential growth of network devices in use, have made multiple, distributed controller setup mandatory. Each controller heads a particular range, domain under it. in this distributed system, controllers need to exchange their domain specific information for effectiveness of the network. Eastbound API is mainly responsible for exchange of information with the distributed controllers.
- Westbound Interface are responsible for, creating a channel between different SDN domain's multiple control planes. These API provide a global view of network, and impacts on routing rules of controllers.few conventional protocols such as Border Gateway protocol can be used between remote SDN domains as westbound interface.
- SDN controller maintains a entire network view under a domain. Manages the infrastructure of the network, and puts northbound API forward for applications. Learning switch, load balancer, routers are some of the normal application that come with SDN controllers. End-user device discovery is one of the features of SDN controller which identifies the end users of the network. Such as laptops, mobile phone, any device with network interface card and with configuration of IP in it. SDN controllers can detect the network devices engaged in network, with help of one more core feature that is Network Device Discovery. It identifies the network devices that are responsible for building the network. Network is nothing but interconnection of devices, these connections possess a particular topology. One more core feature of SDN controller, Network Device Network Topology Management, maintains information of such interconnection details, network topologies as well as information of which device is connected to which end-user device. Flow management is one more very important core feature of SDN controller. controller manages the packet traffic flows. Flow management maintains database of flows. And ensures synchronization.

III. LIGHTWEIGHT IDS

A network consists of many connection points, pre-installed application act as brain at each connection point. SDN is the replacement of all brains with a single controller brain, which is called centralized control or controller. Lightweight IDS are the Application systems which can be easily installed and deployed on any connection point in network. And system should be easily updateable, configurable as well as manageable. SNORT is very less in size compared to that of commercial IDS. SNORT is a packet sniffing tool under libpcap. This lightweight intrusion detection system can be used as a logger. Rule writing mechanism makes rule-based pattern matching possible. stealth port scan, buffer overflow, SMB probe, CGI attack are some attacks detected by SNORT. Snort has alerting mechanism, syslog alerts, server message blocks. It has been configured using BPF. Testing and actions are done on each packet. Snort is also a sniffing tool. Snort shares commonalities with both sniffers and NIDS.In almost all cases SNORT is an easy to handle application, both technically and commercially. Snort is cosmetically almost like tcpdump

but is more focused on the safety applications of packet sniffing. The major feature that Snort has which tcpdump doesn't is packet payload inspection. Snort decodes the appliance layer of a packet and may tend rules to gather traffic that has specific data contained within its application layer. This allows Snort to detect many sorts of hostile activity, including buffer overflows, CGI scans, or the other data within the packet payload which will be characterized during a unique detection fingerprint. one more benifit of snort is that its decoded output is more understandable than that of tcpdump. Snort doesn't currently lookup host names or port names while running, which may be a function that tcpdump can perform. snort is responsible for assembling packets quickly and processing them within the snort itself. Performing run-time host name lookup isn't conducive to high performance packet analysis. Perhaps the simplest comparison of Snort to NFR is that the analogy of Snort as brother to NFR's collegebound football player.

Snort shares a number of an equivalent concepts of functionality as NFR, but NFR may be a more flexible and complete network analysis tool. That said, the small brother idea might be extended therein Snort tends to suit into small places and is somewhat more ''nimble'' than NFR. For example, NFR's packet filtering n-code language may be a serious, full featured scripting language, while Snort's rules are more one dimensional. On the opposite hand, writing a Snort rule to detect a replacement attack takes only minutes once the attack signature has been determined. NFR also features a more complete feature set than Snort, including IP fragmentation reassembly and TCP stream decoding, are the main features that are necessary for any commercial IDS which does critical attack detection, and NFR was the primary product which could defeat anti-NIDS attacks outlined by Ptacek and Newsham. Presently, Snort doesn't implement TCP stream reassembly, but future versions will implement this capability. with the help of snort, fragmentation of ip including the option of rules, and with minimum size threshold for packet fragments are available. The decode engine is organized round the layers of the protocol stack present within the supported data-link and TCP/IP protocol definitions. every subroutine of the decoder imposes order on the data by usind data structure on the network traffic. These decoding routines are called so as through the protocol stack, from the info link layer up through the transport layer, finally ending at the appliance layer. in this section speed has been given importance, and therefore almost all functionality of the system consists of setting pointers into the packet data for later analysis by the detection engine. decoding is provided by snort for ethernet, datalink protocals and serial line internet protocal.

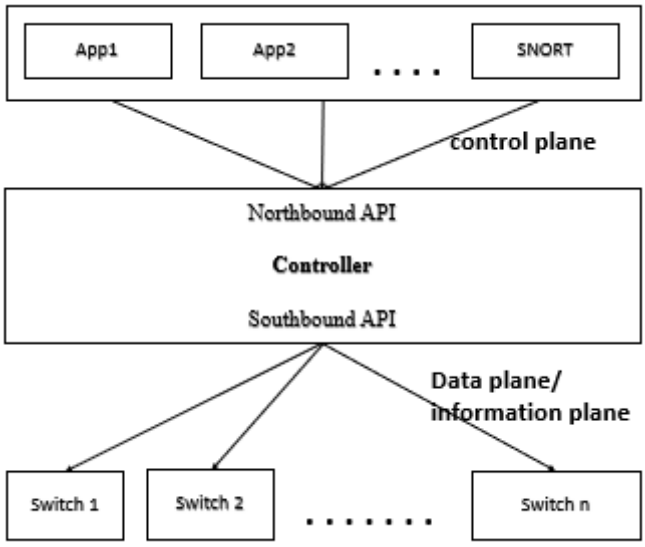

Fig 1. Simple overview of SDN architecture

Snort maintains its detection rules during a two-dimensional linked list of what are termed Chain Headers and Chain Options [7]. These are lists of rules that are condensed right down to an inventory of common attributes within the Chain Headers, with the detection modifier options contained within the Chain Options. For example, if forty-five CGI-BIN probe detection rules are laid out in a given Snort detection library file, they typically all share common source and destination IP addresses and ports. for improving the speed of detection technique, these features are combined into single Chain Header then specific detection patterns are kept in Chain Option patterns. These rule chains are searched recursively for every packet in both directions. The detection engine checks only those chain options which have been set by the rule's parser at run-time. The first rule that matches a decoded packet within the detection engine triggers the action laid out in the rule definition and returns. A major overhaul of the detection engine is currently within the planning and development stage. The next version of the engine will include the potential for users to write down and

distribute plug-in modules and bind them to keywords for the detection engine rules language. Because of this, an appropriate plugin module for featuring significant detection mechanism for snort and particularization of the system for a specific work is possible.

IV. PROPOSED METHODOLOGY

SNORT is a best-known lightweight intrusion detection system. Which has a powerful packet classifier and a real-time alerting mechanism, which can be a great advantage for one who adopts it. SNORT alone is a lightweight intrusion detection system; it cannot be replaced by a commercial IDS. the strength of Snort can be enhanced by collaborating some of the advance features of snort with the implementation. The current paper content revolves around how strength of snort can be enhanced by existing advance features of snort along with rule writing feature of snort. And also, how efficiently the real-time alerting mechanism, and packet classification can be adapted and utilized effectively so that a standalone network intrusion detection system can be made out of it. Using some other existing opensource tools and technics, snort can be enhanced with its detection capabilities. Network applications needs resources, and real time implemented networks for experimenting and implementing new systems developed, which would cost a lot. The current paper is about how to make the SNORT more effective by implementing some effective methods in the SNORT installed in the existing system [6]. The current paper uses the implementation of the system proposed by authors of paper [6]. The SNORT which has been implemented can be more effective in the system if these methods are included in it [7].

- Supporting proprietary intrusion detection systems: attack detection works based on knowledge of previous attack patterns, and vendors are responsible for any new attack pattern updates. If that updates are failed or delays, then the system may become vulnerable for attacks. To cover this gap, SNORT can be used to get the signature of the new attack by executing it on local dummy network. The signature of the obtain new attack will be made as a rule for reference. Berkeley packet filtering can be used to refine the packets analyzed by SNORT.
- Honeypot traps: are systems, or computers that monitor and swindle the antagonistic parties. Honey pots monitor and record the data once they get started. These recorded data can only be interpreted by SNORT. Best analyst for interpreting the honeypots output. SNORT has packet classifier which will be a supporting factor in interpreting the honeypot results. Honeypot with SNORT can be an autonomous intrusion detection system for network. SNORT can detect backdooring, antagonistic parties, and other trespassing in network by writing rule for monitoring the non-existing services, not in use ports. Packets heading towards non-existing ports is nothing but backdooring, or may be an attack.
- Shadow with SNORT: Shadow was developed by US navy as a defense project. Tcpdump is the sensor part which records the traffic to a file. This recorded traffic is analyzed by the analysis station. Analyzed data will be passed through Perl filters for further processing. Shadow doesn't have good packet classifier, as well as alerting mechanism. Tcpdump sensors can be replaced by SNORT, as it has best packet classifier and real-time alerting system.

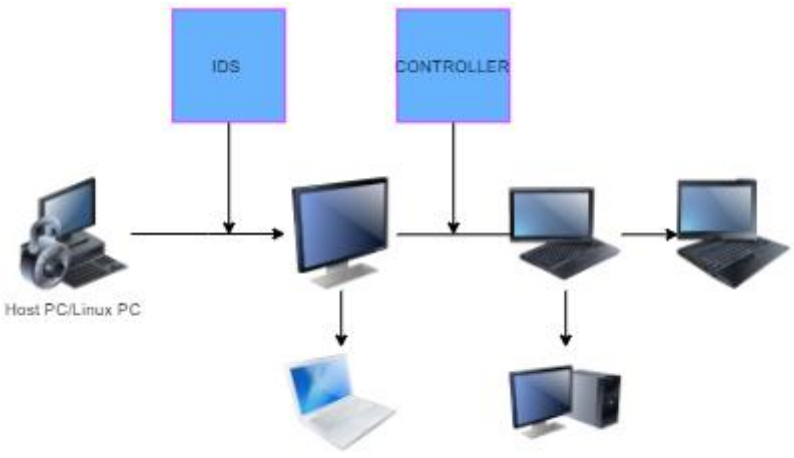

Fig 2. SDN with IDS in firms' network

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

V. SOME FEATURES OF THE SYSTEM

Software-Defined Networking has some major features, which it focuses on: control plane of network and routing or forwarding planes are completely disintegrated. The network activity segregation is done. Forwarding and filtering, implemented in tables of hardware they remain as it is in the device. The main thing, software which is responsible for controlling the device and as well as the packet traffic flow, is removed from device and placed into centralized controller. The isolation of control software from device to centralized controller has made the complete view of the network available for the control software. Optimal routing and traffic forwarding decisions are now possible.
The network forwarding is abstracted from hardware to asoftware layer which is programmable. An entity, known as a controller, is introduced to coordinate and manage, network-widetraffic forwarding decisions, Intrusion detection is one of the main issues for the network security.

VI. CONCLUSION AND FUTURE WORK

Network virtualization came into existence by SDN, which made network managers to a handle networks with a centralized control approach. There are many existing solutions for intrusion detection and prevention, but for Software Defined Network, only few. SNORT is a lightweight IDS system, which shares some common features from tcpdump. The current work is to show how SNORT can be implemented as IDS for SDN. Collaborating SNORT with some existing mechanisms will yield standalone NIDS, efficient than commercial NIDS. In this paper, existing work [6] is re-implemented to enhance the performance of SNORT with some added features. Collaboration of SNORT with SHADOW, Honeypot, snort can be a best alternative for proprietary IDS systems.Software defined networking will be reliable for implementing an Intrusion detection System. asa future work, the non-functional requirements will be considered to achieve more reliability and robustness and efficiency, with many other applications and IDS running over the network. The defensive mechanism can be strengthened by understanding some more malwares, and attack patterns. Future work will be on learning new malware patterns and attacks.

REFERENCES

1. https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html
2. Tella Nagarjuna Reddy, K.AnnapuraniPanaiyappan, “Intrusion Detection on Software Defined Networking” SRM Institute of Science & Technology, International Journal of Engineering & Technology 7 (3.12) (2018) 330-332
3. T. Xing, Z. Xiong, D. Huang and D. Medhi, "SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds," 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, 2014, pp. 308-311, doi: 10.1109/CNSM.2014.7014181.
4. H. Hendrawan, P. Sukarno and M. A. Nugroho, "Quality of Service (QoS) Comparison Analysis of Snort IDS and Bro IDS Application in Software Define Network (SDN) Architecture," 2019 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 2019, pp. 1-7, doi: 10.1109/ICoICT.2019.8835211.
5. R. M. A. Ujjan, Z. Pervez and K. Dahal, "Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN," 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Island of Ulkulhas, Maldives, 2019, pp. 1-8, doi: 10.1109/SKIMA47702.2019.8982413.
6. Pavithra H, Abhishek A, Jayachandra M, Punithraj M N, Umakanth H P, “Signature-Based IDS for Software-Defined Networking”, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 7, Issue 9, September 2018, ISSN: 2278 -7798.
7. Martin Roesch, “SNORT-LIGHTWEIGHT INTRUSION DETECTION FOR NETWORKS”, page number 234, Proceedings of LISA '99: 13th Systems Administration Conference Seattle, Washington, USA, November 7–12, 1999.
8. Z. Zhou, Chen Zhongwen, Zhou Tiecheng and Guan Xiaohui, "The study on network intrusion detection system of Snort," 2010 International Conference on Networking and Digital Society, Wenzhou, 2010, pp. 194-196, doi: 10.1109/ICNDS.2010.5479341.
9. R. Gaddam and M. Nandhini, "An analysis of various snort-based techniques to detect and prevent intrusions in networks proposal with code refactoring snort tool in Kali Linux environment," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017, pp. 10-15, doi: 10.1109/ICICCT.2017.7975177.
10. A. Garg and P. Maheshwari, "Performance analysis of Snort-based Intrusion Detection System," 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2016, pp. 1-5, doi: 10.1109/ICACCS.2016.7586351.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

11. Z. Kai, "Research and Design of the Distributed Intrusion Detection System Based on Snort," 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp. 525-527, doi: 10.1109/ICCSEE.2012.310.
12. S. Kumar and R. C. Joshi, "Design and implementation of IDS using Snort, Entropy and alert ranking system," 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies, Thuckafay, 2011, pp. 264-268, doi: 10.1109/ICSCCN.2011.6024556.
13. Sebastian Seeber, Lars Stiemert, Gabi DreoRodosek, "Towards an SDN-Enabled IDS Environment", Neubiberg, 85577, any, 978-1-4673-7876-5/15/ ©2015 IEEE 751.
14. AkramHakiria(b), Aniruddha Gokhale(c) , Pascal Berthoua,(b), Douglas C. Schmidt(c) , Gayraud Thierry(a,b), "Software-defined Networking: Challenges and Research Opportunities for Future Internet", (a)CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France (b)Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France (c) Institute for Software Integrated Systems, Dept of EECS Vanderbilt University, Nashville, TN 37212, USA
15. https://www.sdxcentral.com/networking/sdn/definitions/southbound-interface-api/