Computers, Materials & Continua CMC, vol.62, no.3, pp.1077-1096, 2020

Ameliorate Security by Introducing Security Server in Software Defined Network

J. Vijila[1, *] and A. Albert Raj[2]

Abstract: Software Defined Network (SDN) deals with huge data processing units which possess network management. However, due to centralization behavior ensuring security in SDN is the major concern. In this work to ensure security, a security server has been at its aid to check the vulnerability of the networks and to keep an eye on the packet according to the screening policies. A Secure Shell Connection (SSH) is established by the security server which does a frequent inspection of the network's logs. Malware detection and the Intrusion Detection System policies are also incorporated in the server for the effective scanning of the packets. In response to a suspicious log or the packets in the SDN network there is a change in the security norms. Hence the proposed work updates the security policies in accordance with the attacker mentality.

Keywords: SDN, security server, malware deduction.

1 Introduction

In the recent track, the world is more comfortable with the elevation of upcoming technologies notably cloud computing, big data and IOT. These preferences work on the basic of information. This insists the importance of Network security in the modern era. Traditional implementation of network security has a single point of security check named as firewall [Ahmad, Namal, Yilanttila et al. (2015)]. Earlier, the firewall was well and good. But when the time changes it cannot run behind the recent advancement in the techno world. SDN is a new network management model that solves most of the problems of old networking models. SDN is designed to decouple the controller from the data plane in the switches that provide intelligence in the centralized controller [Ahmad, Namal, Yilanttila et al. (2015)]. The controller is acting as a main controller for making routing decision which is responsible for making routing decision using OpenFlow protocol. One of the research fields in SDN is providing overall security. In SDN there are two levels of security: Providing security to Secure Socket Layer (SSL) and providing security to devices such as switches, servers and end devices. The Open Flow gives a tool to secure the various devices. At this point when security is not providing, a malicious device can imitate and acted as a real device. SDN suffers from variety of security issues

[1] University College of Engineering, Nagercoil, 629004, India.

[2] Sree Krishna College of Engineering and Technology, Coimbatore, 641008, India.

[*] Corresponding Author: J. Vijila. Email: vijilaebenezer@gmail.com.

such as DOS (Denial of Service) attack, Structured Query Language (SQL) injection and malware threat [Cui, Karama, Klaedtke et al. (2016)]

The traditional networking, architecture is built upon the foundation of hardware components such as router, switches, host machines, servers etc. To accompany them there are a number of protocols for the effective functioning of the networking. Over time the network has become larger and the necessity of the high rate data transmission is motivated by the recent advancement [Sharma and Tyagi (2018)]. The competitive market vendors are keeping their race by meeting the requirement of the current network by introducing new protocols such as OSPF (Open Shortest Path First), Border Gateway Protocol (BGP) and so on. After every serious advancement, the networking issues and complexity have not come to the end, because the root cause for the entire problem lies in the foundation of the network [Sharma and Tyagi (2018)]. Fig. 1 shows the casual networking architecture with LAN and WAN connections along with the firewall to screen the unwanted traffic from the external packets. But adding a single check point is not enough to filter all the threats.

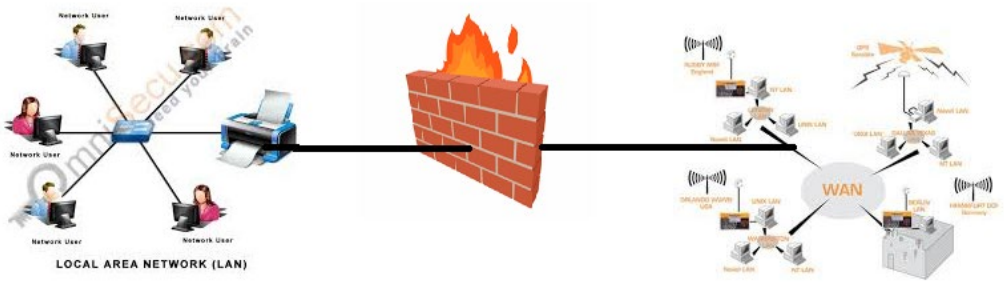

Figure 1: Traditional architecture of Firewall

Hence, SDN has found a solution for it by evolving the centralized controller and the programmatic switches. The idea of SDN is simple and precise. It focuses on changing the control from the hardware component to the software component. Technically speaking, in a SDN, a network engineer or administrator shapes traffic from a centralized control console without having to touch individual switches in the network. The centralized SDN controller maps the switches to forward the data packets (network services) wherever they are needed, no matter what are the specific connections between a server and devices. But this single checkpoint is not enough to filter all the threats.

In SDN the control plane from various switches has been decoupled from the data plane and projects as the single controller machine. This process is a one step ahead process away from traditional network architecture, in which individual network device makes traffic decisions based on its configured routing tables. Hence in this work, to enhance the security of SDN network and to reduce the controller overhead a new security server is added in the SDN architecture. The security server which acts as a brain of the SDN gives access to modify the network traffic and avoids access to the legitimate users (DOS attack) and thus provides higher security when compared to existing networks.

The rest of this paper is organized as follows. In Section 2, potential security issues

corresponding to this research are analyzed. Section 3 presents the security architecture of the proposed work. Section 4 and Section 5 projects the implementation and Result analysis of the proposed work and finally conclusion is addressed in Section 6.

2 Literature review

Many security algorithms for SDN are proposed in the literature. Lara et al. propose the implementation of the programmatic switches using T-CAM and RAM to analyze the traffic and improve the performance of the network [Lara, Kolasani, Ramamurthy et al. (2013)]. Cui et al. [Cui, Karama, Klaedtke et al. (2016)] use the practicing of fingerprint attacks using RTT and Packet dispersion for addressing the active and passive attack inorder to safeguard the switches [Cui, Karama, Klaedtke et al. (2016)]. Ahmad et al. [Ahmad, Namal, Yilanttila et al. (2015)] develop ethane security using SANE for addressing the various security issues and counters. Tomar et al. [Tomar and yagi (2014)] discuss the various threats in network and their counter measures for providing the awareness in the large scale and small scale networks. For securing end-host attacks, the authors in this paper propose a technique for validating the switch state in the data plane. A feature extraction method vector of n-queens is added by protecting the switches and controller from attacks [Tupakula, Varadharaja and Mishra (2019)]. The authors propose a controller that is extensible and easy to modify in NS-3 simulator with OpenFlow protocol [Ponvea, Pistirica, Moldoveanu et al. (2019)].

Jankowski et al. [Jankowski and Amanowwicz (2016)], use a self-organizing map for detecting intrusion. The proposed approach detect U2R Attacks by including packet inspection technique Shahzad et al. [Shahzad, Mujtaba and Elahi (2012)] propose and discuss about the southbound interface, forwarding devices and the management plane. The authors Yao et al. [Yao, Han, Sohail et al. (2019)] in this paper propose SDN- based 5G networks for providing security in SDN network by adding extra cryptographic authentication and a synchronization secret.

Singh et al. [Singh and Guntuku (2014)] use open source tools by adding machine learning approach for providing security. Li et al. propose an efficient SDN controller scheduling algorithm for handling Distribute DoS (DDoS) attack [Li, Cui, Ni et al. (2019)]. Chen et al. provide a practical solution to collect and analyze data in cloud computing platform for finding various malicious attacks [Chen, Han, Cao et al. (2013)]. Giotis et al. [Giotis, Argyropoulos, Androulidakis et al. (2014)] combines openflow and SFlow for detecting the various anomalies on SDN environment.

Even though, a lot of mechanism is cited in the literature, the attack in the control layer creates an adverse effect, since it takes care of the controller which acts as a brain of the SDN. Attacking control layer gives access to modify the network traffic and avoids the access to the legitimate users (DOS attack). To reduce the threat and to avoid the malware packets problems, a three-layer framework is developed by adding a security server to enhance the security of SDN network and to reduce the controller overhead.

3 Proposed system

In the proposed work, SDN architecture is established with a single controller configuration along with a switch, host and the routers. In addition to the traditional architecture, this novel approach uses one security server to take care of the packets. Fig. 2. shows the architecture of the proposed work with a single controller configuration. In the infrastructure layer, the OpenFlow switches are implemented to maintain a secure communication between the switches and the controller and three host machines are connected to the OpenFlow switches at the bottom of the configuration. The control layer deals with the administration management of the network with the implication of controller. The packet transmission between the controller and the security server takes place through the routing protocol such as OSPF (Open Shortest Path First) and the authentication dosage is controlled by the TACACS. In the application layer, the services are published by the end users.

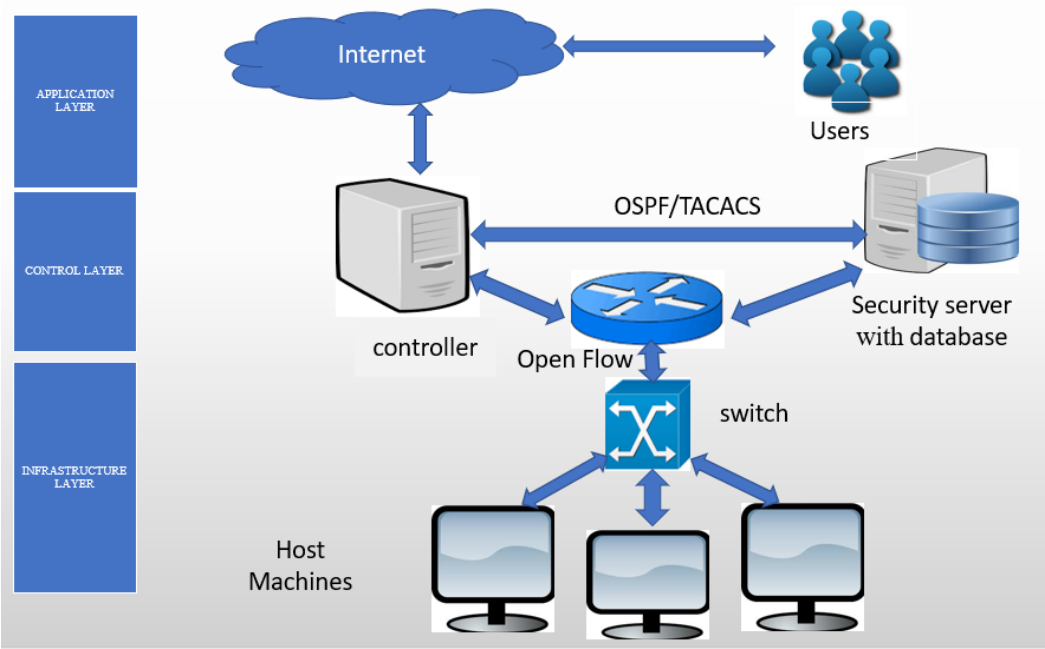

Figure 2: Software defined network architecture of proposed work

3.1 Security server

The scalability, reliability, consistency, and interoperability of the controller lie in the hands of efficient design of SDN network with the reliable changes in the architecture. The security server solves many of the attacks mainly the Denial of Service (DOS) attack. Additionally, the attackers gain full control over the network once they configure the controller, since they act as the single point of entry. So, to update the security concern issues, in the SDN, the architectural design is proposed by introducing the security server in the existing architecture. This is illustrated in the Fig. 3.

In Fig. 3, the security server has a remote system logging option using the SSH protocol

and also it is incorporated in the SFTP for file transfer. The server is fused with the various up gradation tools under the platform of kali Linux. The admin is allowed to check the network with the accessible penetration testing tools which are pre-configured in the system. The authentication of the admin is done by using the username and the password.

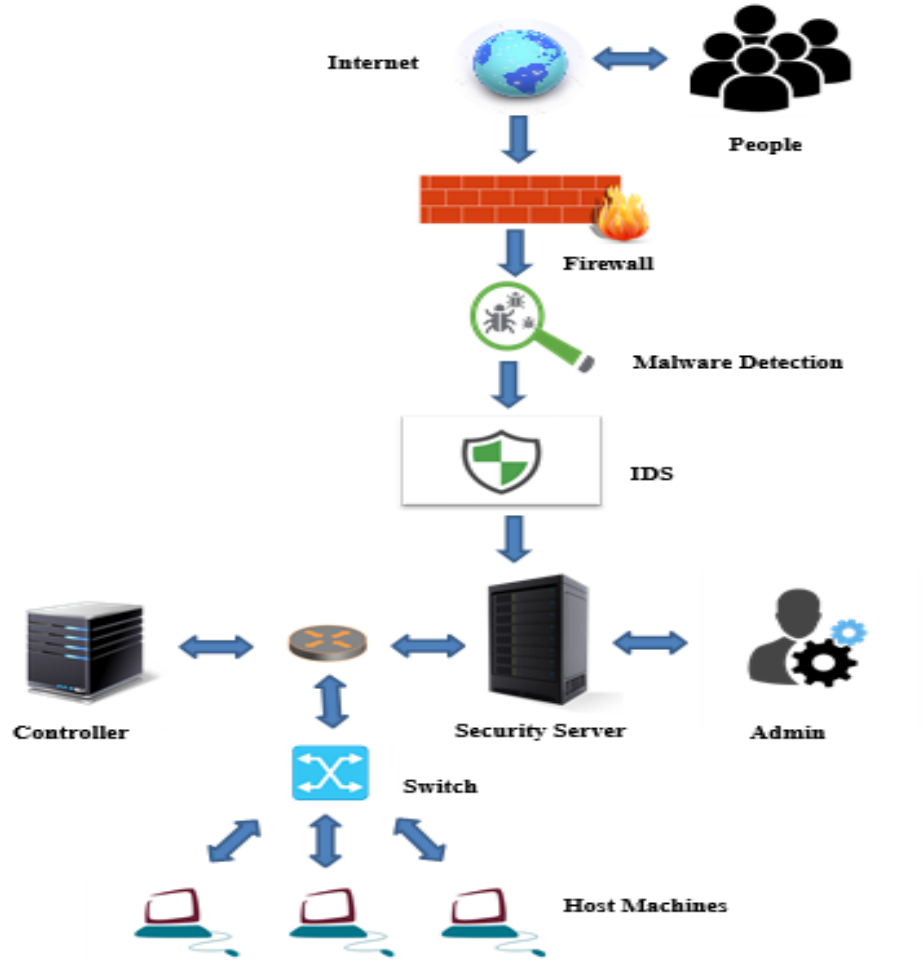

Figure 3: Detailed architecture of proposed system

Now the packet reaching the security server is filtered using the Intrusion Deduction System (IDS) that is incorporated in the system. The IDS keeps an eye on the incoming packet and discards the packet with infrequent pattern. Another payload to ensure security is that the malware analysis which checks for any rootkit in the network and works for the back rolling of the packet acts before the Intrusion Deduction System.

3.1.1 SSH connection

The Secure Shell Connection between the server and the admin has the prior security concern by establishing the secure connection. The ideology of introducing an admin for

the purpose of security server handling is that the network topology changes at any concern and also the attack is not the same at every point. Since there is an advancement in the technology, the attackers and the attacking platform also have the enhancement along with that. So, the pre-planned security policies are not always suited and are ready to the upcoming updates. So, the server requires a man's help to upgrade and update it. Since it has done its best to suit the SDN network it also has some misleading traffic or packet in them.

The mandatory task of the admin is to check the network frequently for any mismatching or infrequent history. Here the admin refers to the security engineer of an organisation or the institute where the SDN in implemented and gets a look over of the logs in the network by penetrating the network and exploring any mismatch frequencies. The effectiveness of the SDN network lies in the intelligence of the admin. Once the suspected act is encountered, the network is reframed by finding out the affected file and changing the permission of the concerned part. This in turn pushes the admin to reframe the security policies in order to patch up the loop holes in the network. The pen-testing tool is preconfigured in the security server as an aid to the admin.

The security server is updated periodically in order to keep it away from the inside intruders. Setting up the file permission, changing the framework policies, configuring the updated security frameworks, upgrading the pre-installed tools, setting up the new secure Application Programming Interface (APIs), discarding the packets that are ruining the network beyond the security policies are the packages that should be noted by the admin in the security server as well as in the network.

3.1.2 Malware detection

Malware refers to the malicious software that is downloaded or entered into a system with the intention of attacking an individual system or the whole network connected of that system. This is a major threat if it goes unnoticed. It deploys the mechanism of encrypting the files in the system where it is downloaded and further configures the network by demanding a favour to decrypt the files in the network. It is popularly known as the Ransom malware and it has shown its strength in the recent times. The other types of malware are worms, rootkit, virus, key logger and the Trojan.

To cope up with these malwares there are many malware detection software that play its effective role in the detection of malware. This software is incorporated into various logics and policies to get rid of the malware in the security server. The admin keeps an eye on the malware detection software and updates it on a timely basis. In the pursuit and development of the malware detection algorithms, often a big sample set of both malicious and benign samples are required. Machine learning or similar automated techniques, as well as manual or partially manual signature generation, often requires a good and varied example set of benign samples that are commonly mistaken as malicious. Those samples are usually analysed automatically and then given to a reverse engineer for further scrutiny, analysis and improvement of said malware detection algorithm.

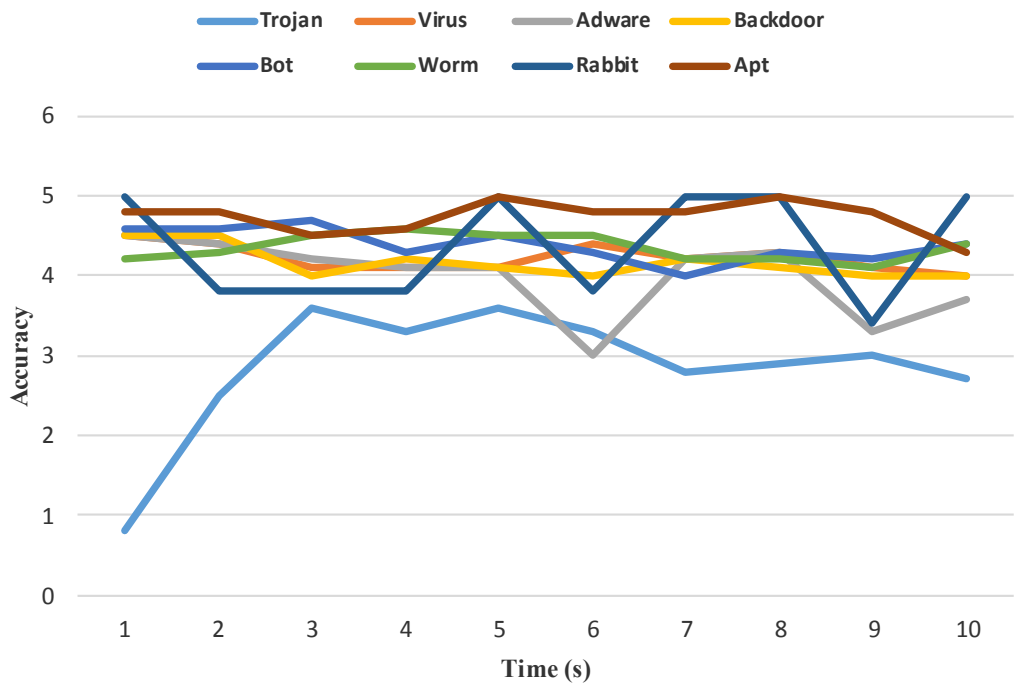

Figure 4: Malware Detection report on various families of threat

Fig. 4 points out the graphical fact of the malware detection software that work on the basics of the neural network. The graph has been focusing on the detection key of the malware software with respect to the time and the accuracy. Eight families of malware threat have been preconfigured with this novel implication.

3.1.3 Intrusion detection system

Intrusion Detection System (IDS) is a software application that detects the malicious threat in the system and sends the report to the admin that is collected centrally using the security server. The configuration deploys a SIEM system that collects the output from various sources and uses alarm filtering techniques to distinguish malicious activity from the false alarms. IDS manipulates the prevention mechanism by primarily focusing on the identification of the possible incidents such as logging information, reporting attempts etc. It keeps on track every event in the security server by spying the logs and the in and out movement of the packets in the security server system.

The ideology to introduce the IDS in the security server is to take over the disadvantage of the firewall. Firewall limits access between the networks, since it prevents the intrusion but it does not signal an attack from inside the network. It is dumped with the history of various known patterns of attacks and heuristic for the intension of scanning the malicious threat in the system. This pattern of attacks and heuristic will be updated periodically according to the intelligence of the admin. The scanning report is submitted to the admin for reference. According to the reporting perception, the network is remodeled or refigured and the security concerns are ensured with the double way sequence.

Fig. 5 projects the flow and control of the data over the network. The data is flowing from outside of the network to the internal SDN network in a sequential manner. First the packet has a flow entry from the upper external layer that is unaware of the internal network. Then the packet is routed to the controller in the SDN network and the security trails are carried out in the security server for labelling their source or the origin of packets and to check whether the packets are from known or verified sources.

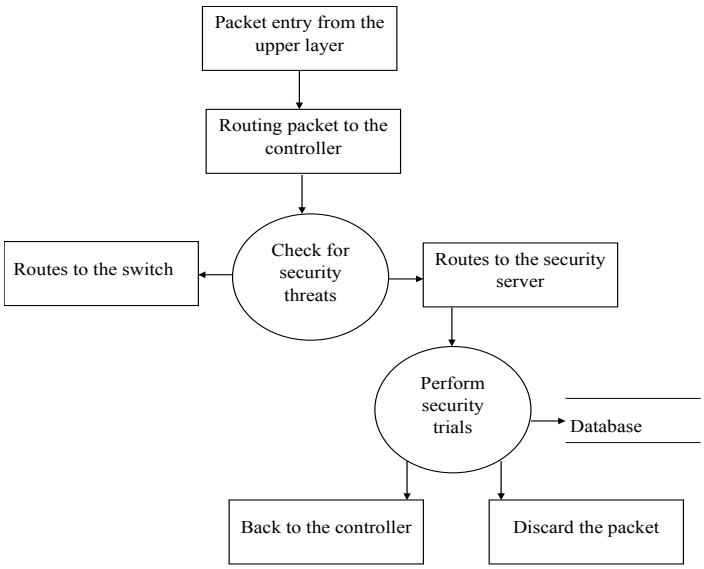

Figure 5: Data flow diagram of the proposed work

4 Implementation of the proposed work

The enforcement of this ideology is carried out in a three-step module for the purpose of easy handling. The foremost step to accomplish the task is to set up an SDN network along with the security server. The next module is packaging up the tools and software required by the security server for its start-up. Along with the secure SSH connection is established between the client and the Admin.

4.1 Establishing an SDN network

The SDN network outlines the distribution of single controller across the host machine of number three. This is congregated in the mini-net which is a software that virtualises the SDN network and tests the design across various performance ratios. Fig. 6 shows the SDN architecture setup in a linear fashion of three host machines h1, h2, h3 connected to the switch: s1, s2, s3 under the single controller configuration of c_0. The host machines are configured by creating a link between the h1→ s1, h2→ s2, and h3→ s3. The three switches in the network are connected in a linear manner of s1→s2→s3. Now the controller c_0 is connected at the top of the network by connecting all the three switches to it. The reachability to each node in the network is tested and surveyed. The packets are routed to every host machine in the network and also the reachability of each and every

host machine is tested from the h1→h2, h3 h2→h3, h1and h3→h2, h1. The result is captured with 100% throughput with no packet loss.

4.2 Configuring the security server

The next step is absolutely setting up an pentesting laboratory with all its upgraded tools as a security server under the platform of the kali linux. The access rights are given to the admin with the two step authentication purpose of username and password. Fig. 7 represents the failure in the logging process in case if it fails to enter the right user credentials. Once the authorised admin with the authenticated information is encounerted, then the system is configured by the admin. Now the system is updated on a timely basic to avoid any backlogs or the security lags. So it its the prime duty of the admin to update the key of the system. On having a log over the updated app will help the system to run half away against the security threats.

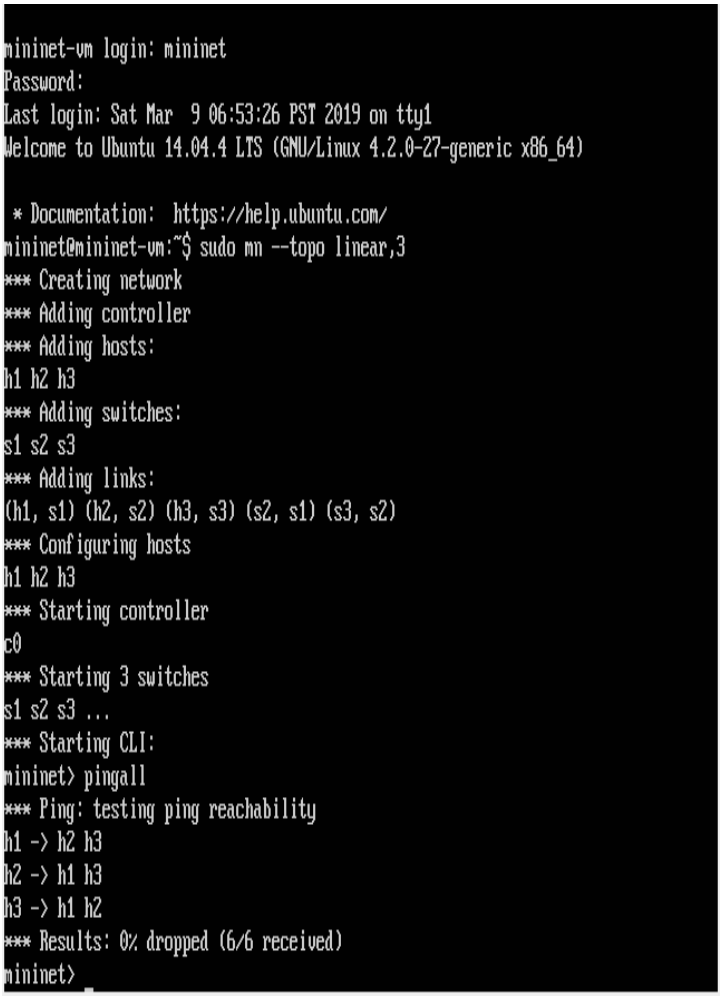

Figure 6: SDN network organization

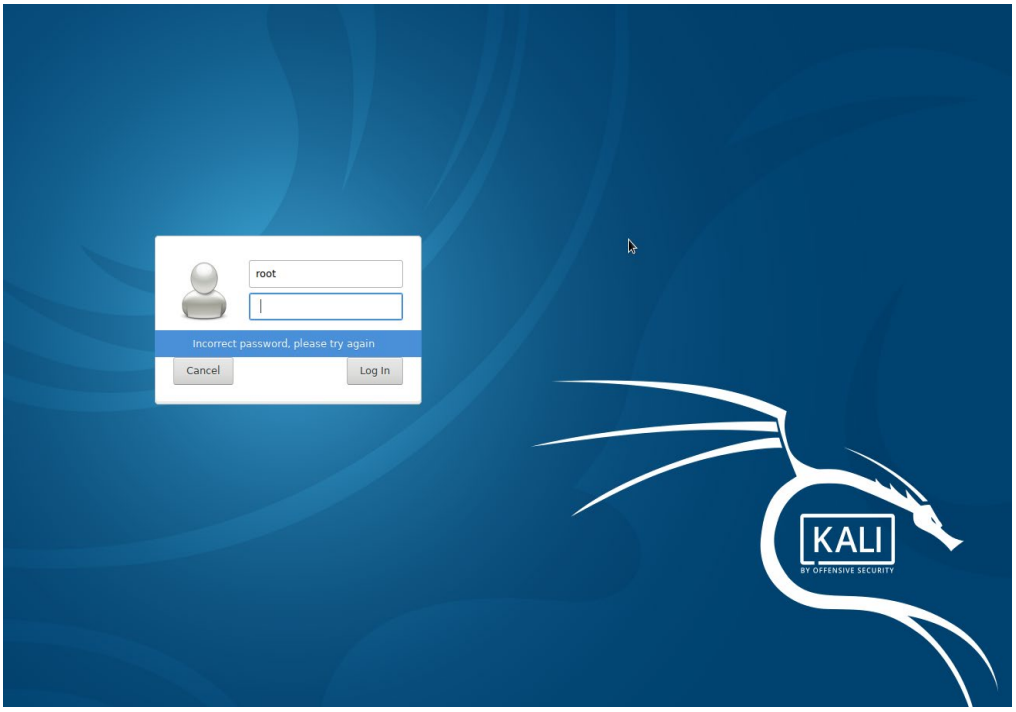

Figure 7: Admin login setup

Fig. 8 pin points the activities of the admin in pre-state configuration of the server for its first time use and the upgradation of the tool in the system. The security system server has to be updated in order to get rid of the outdated threats and malware. Then the status of the security server is checked to ensure the process. The status of various modules such as,

- Apparmor–Application Armor is a Linux kernel security module that is used to restrict the access to the programs in the system
- Console setup.sh-consists of functional keyboard packages for the Debian based platform.
- Cron–operating system command to execute a job at a current time
- Cryptdisk–cryptography setup for the disk partition
- Cryptdisk-early–Holds the information about the early stage cryptography details of the disk partition
- D-Bus–software bus used for communication among multiple computer programs
- Hwclock–command used for checking the status of the system clock
- Irqbalance–hardware interrupts the works across the multiprocessor to increase the performance of the system
- Keyboard setup.sh–configures the keyboard access to help the system services

- Kmod–user space module used to manage the Linux kernel module
- LightDM–display module in the Linux based system, which is an opensource and cross platform
- Live-tools–used to check the status of the live session tools in the security server
- Network Manager–daemon that sits on the top of libudev and another Linux kernel interface to provide a high-level network interface.
- Networking–commands to look upon the network setup such as IP address port number etc.
- Pppd-dns–to check the status of the dial-up modems
- Procps–bunch of packages that give the information about the process
- Rsync–backup and mirroring command that uses quick algorithm for transferring the files
- rsyslog–logging manager for the Linux based system
- Ssh–cryptography protocol to securely operate the network services
- Sudo–program for running the services over the privileged users
- Udev–device manager for Linux based operating system
- Uuid–Universally Unique Identifier used for identifying the disk location
- Xxl-common–colour theme with elegant button

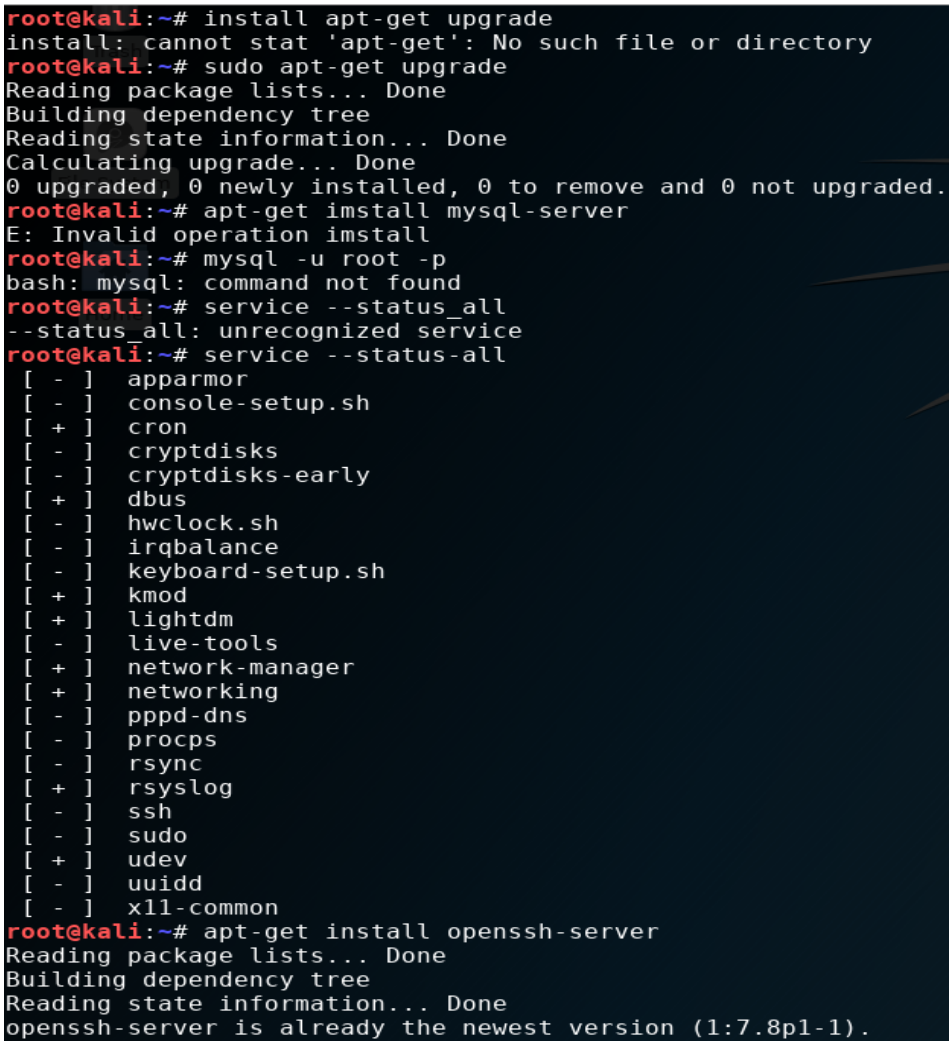

Figure 8: Upgrading the server

4.3 Configuring the secure connection

Once the SDN and the security server is established, the next prior step is to make the connection between the admin and the security server in a secure manner. Still the security constrain should not limit the comfort and the flexibility of the admin. Hence the secure SSH connection is established between the admin and the security server for the remote setup login. Fig. 9 shows an active SSH connection which is established between the security server and the admin. The library files are loaded into the system to meet the requirement of the SSH connection.

- Loaded–tells about the location of the files that are loaded into the system

- Active–portrays the active running state of the SSH server along with the time and date i.e., 12:23:48 and Tue 2019-02-26
- Process–load the various initial process required for the SSH in the executable ID of 1477
- Main PID–notifies the process ID of the active SSH connection. Here the process ID is 1478

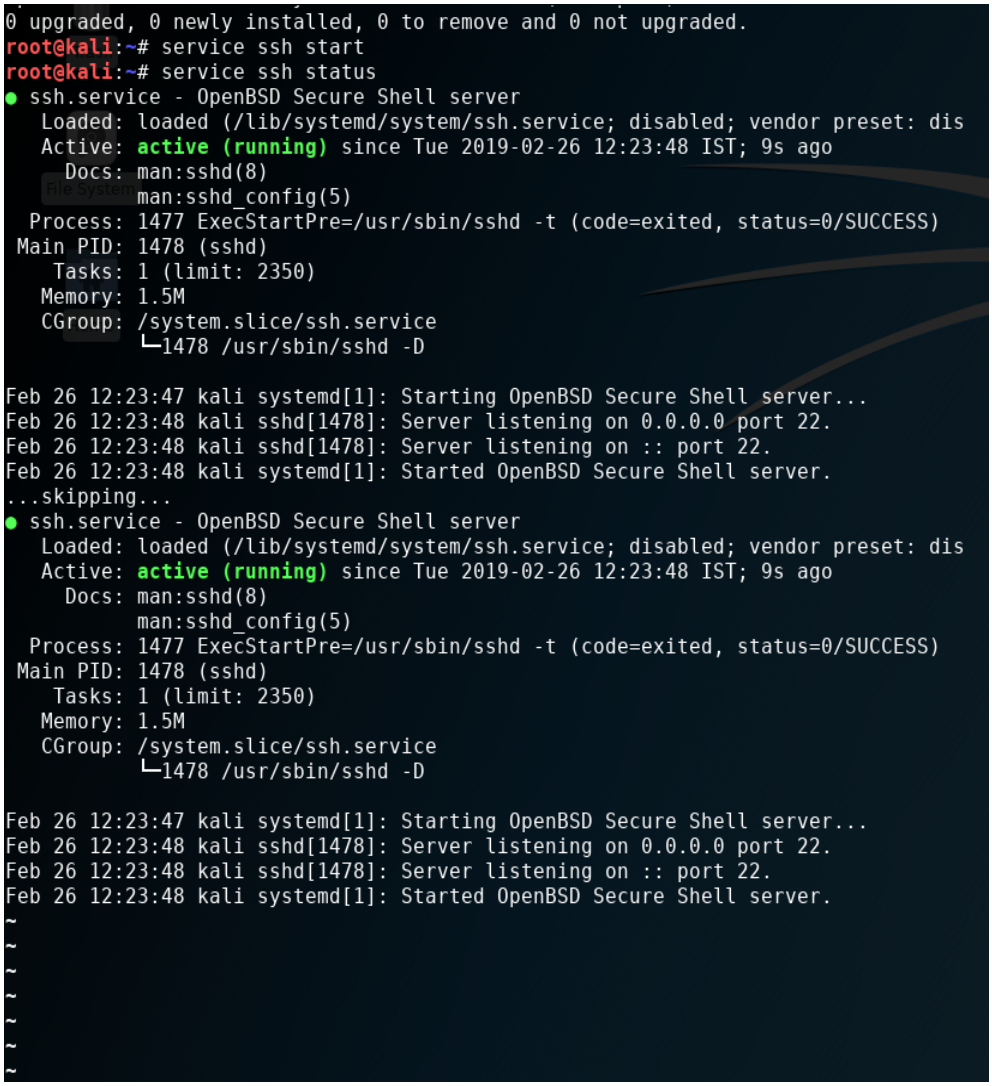

Figure 9: Active SSH connection

Fig. 10 shows the details of the possible sequence of commands for the various operations in the moving task. These commands are used as a manual for the admin. The admin uses this extension for future reference and has a familiar handshake with the platform.

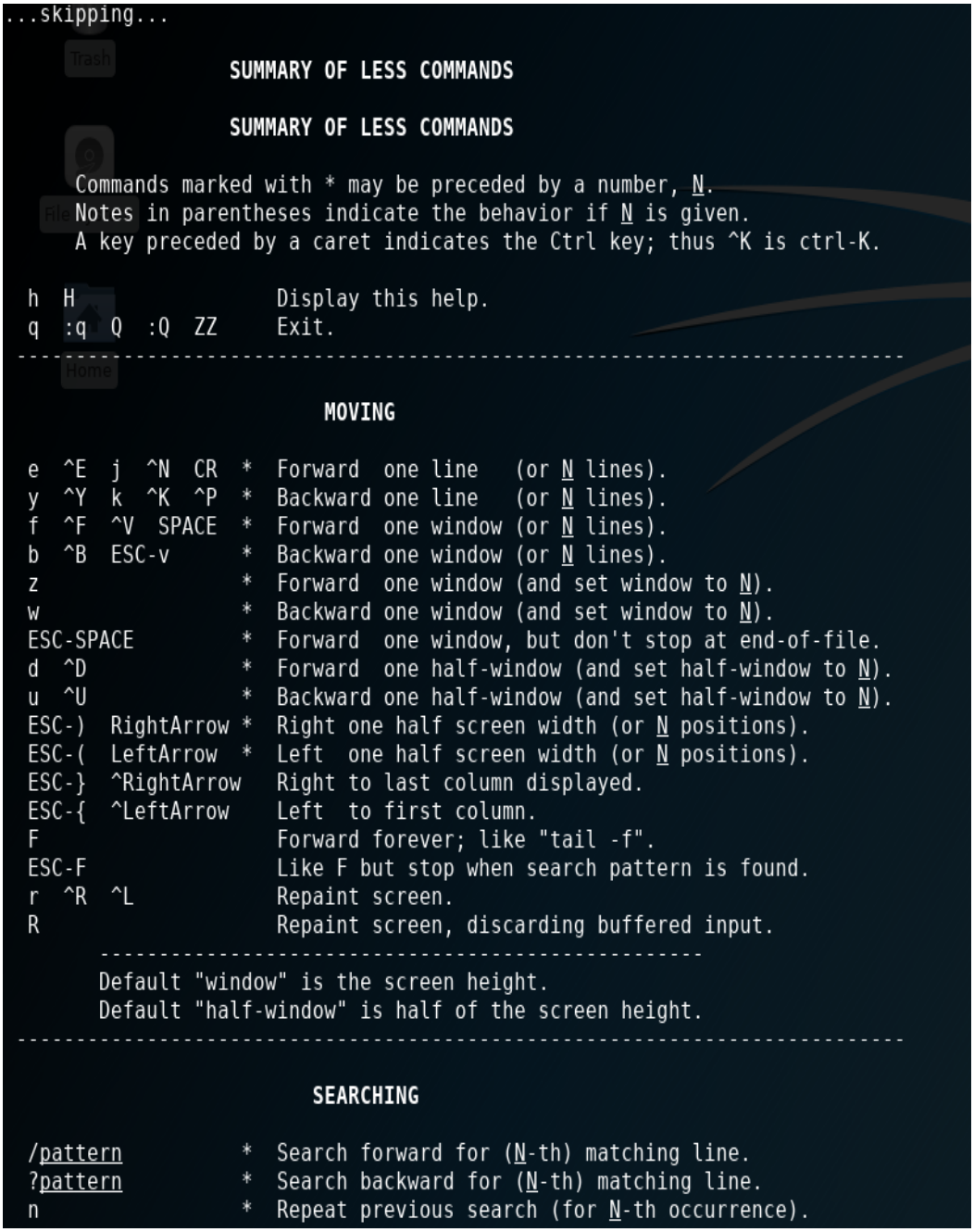

Figure 10: Commands for moving sequences

Fig. 11 configures the various Searching and Jumping command along with the impact of the command on their usage. Thus, the admin takes note on the commands and goes handy with those usages. These are the extension of the above output.

Figure 11: A user manual for the searching and jumping commands

Fig. 12 portrays some of the additional commands involved in the operation of changing files. It also gives an overview of some of the miscellaneous outfits and their functionalities.

Figure 12: Overview of miscellaneous and changing files commands

Fig. 13 specifies the state of active SSH connection and its library files configuration that is established in a successful manner. It also maintains a log for the connection setup in a

detail manner. This acts as an information guide to the admin about the activated SSH connection.

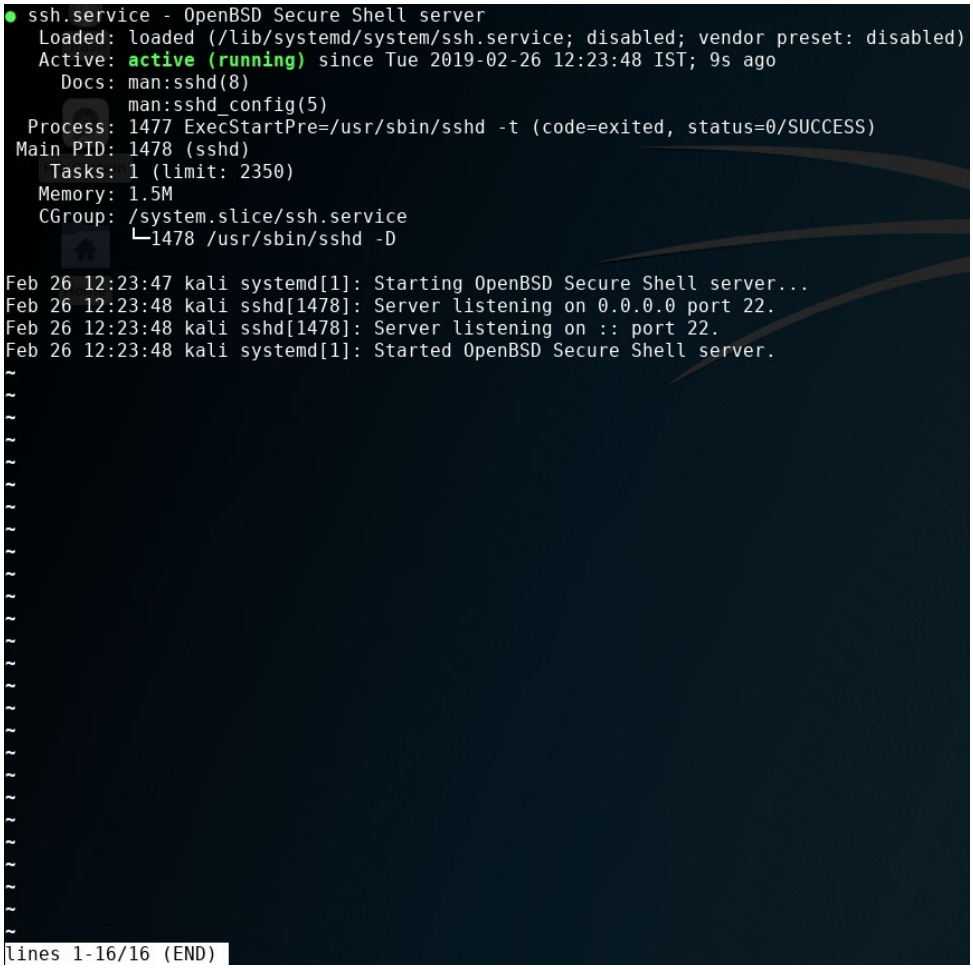

Figure 13: Active specification of the SSH connection

5 Result analysis

The various metrics of the proposed system are cross checked with the traditional and the existing SDN platforms. The results show the desire mannerism with some inevitable bandwidth usage. These records are represented in a graphical or statistic manner for a better reference. It is noted by taking account of three parameters such as security, bandwidth and throughput.

5.1 Performance of security analysis

In security aspects, the proposed analysis shows a higher degree of constrains and accomplishes a three-step check point for the incoming packets. The packets are answerable to the various security trails in the security server. In the traditional network, the firewall acts

as a security platform for the screening of packets. Since firewall is the additional security credential outside the network along with the SDN controller which implies some of the security favours inside the networks, the proposed system gets a third step notification before the controller involvement and does, its best to avoid any threat in the network. Hence the proposed system shows its higher degree of favour to the security constrains.

Fig. 14 discusses the statistic gratitude of the security parameter over the traditional, SDN and the proposed system. It is evident that the proposed system shows the higher degree of security rate when compared to the former. The scale between the time and the attacks shows a gradual improvement when compared to the existing system. Even though the time grade reaches to the maximum of 8 seconds the system is less prone to attacks.

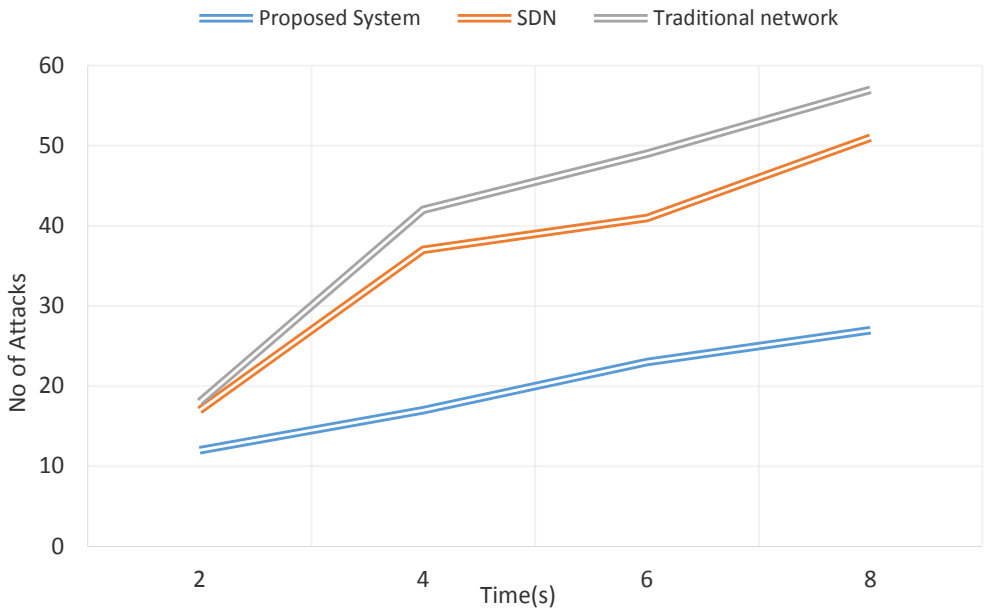

Figure 14: Graph over the security constrain

4.2 Performance of bandwidth

The bandwidth parameter does not match with the desire. Since the proposed environment has to compromise with the bandwidth constrain to achieve a better performance. This is the impact of the fact that some of the overweighed information that are to be routed to the in and out of the network. The existing system shows a favour over this parameter since it does not have some additional or complex system configuration. The traditional outfit does not have any extra configuration so it shows higher degree of bandwidth conservation.

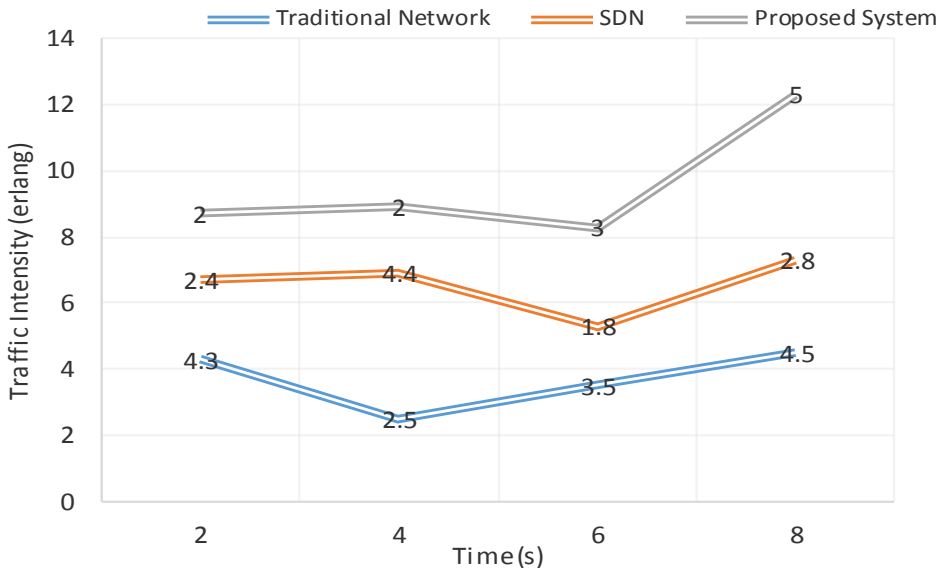

Figure 15: Graph over bandwidth constrain

Fig. 15 deals with the peak usage of bandwidth by the proposed system on a measuring scale of traffic intensity against the time. Along a minimal time period of 2 seconds the traffic intensity reaches a height of 4.2, which impacts in the bandwidth that is inevitable due to the additional implication of security server in its architecture. However, SDN also suffers from the same bandwidth usage when compared to the traditional network due to the implementation of controller in its architecture.

6 Conclusion

Since the security server has done its peak level performance to deploy an effective SDN platform, it is a time variant and time-consuming process by getting on to with various trails and security risks. It comes up with the strategy of trial and error method. But it has a higher degree of security barrier and deducts any kind of security rituals. Hence the pen-testing tools are exploited with the purpose of providing security to the ideology of "If we put ourselves in the attacker's shoes, we might be able to spot a weakness to exploit". In future, we can harden that weakness ahead of time by including various security measures. The bandwidth used by this architecture is looked ahead by framing a new evolutionary techniques or design models.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Ahmad, I.; Namal, S.; Yilanttila, M.; Gurtov, A. (2015): Security in software defined networks: survey. *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317-2346.

Chen, Z.; Han, F.; Cao, J; Jiang, X; Chen, S. (2013): Cloud computing based forensic analysis for collaborative network security management system. *Tsinghua Science and Technology*, vol. 18, no. 1, pp. 40-50.

Cui, H.; Karama, G. O.; Klaedtke, F.; Bifulco, R. (2016): On the fingerprinting of software-defined networks. *IEEE Transactions of Information Forensics and Security*, vol. 11, no. 10, pp. 2161-2173.

Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V. (2014): Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Computer Networks*, vol. 62, no. 2, pp. 122-136.

Jankowski, D.; Amanowwicz, M. (2016): On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. *International Journal of Electronics and Telecommunications*, vol. 62, no. 3, pp. 247-252.

Lara, A.; Kolasani, A.; Ramamurthy, B. (2013): Network innovation using open flow: a survey. *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 1-20.

Li, S.; Cui, Y.; Ni, Y.; Yan, L. (2019): An effective SDN controller scheduling method to defence DDoS attacks. *Chinese Journal of Electronics*, vol. 28, no. 2, pp. 404-407.

Poncea, O. M.; Pistirica, A. S.; Moldoveanu, F.; AsaveI. V. (2019): Design and implementation of an open flow SDN controller in NS-3 discrete event network simulator. *International Journal of High Performance Computing and Networking*, vol. 14, no. 1, pp. 17-29.

Rhode, M.; Burnap, P.; Jones, K. (2018): Early-stage malware prediction using recurrent neural networks. *Computers and Security*, vol. 22, no. 5, pp. 578-594.

Shahzad, N.; Mujtaba, G.; Elahi, M. (2015): Benefits, security and issues in software defined networking (SDN). *NUST Journal of Engineering Sciences*, vol. 8, no. 1, pp. 38-43.

Sharma, P. K.; Tyagi, S. S. (2018): Strengthening network security: an SDN (software defined networking) approach. *ICSCAAIT*, pp. 78-82.

Singh, K.; Guntuku, S. C.; Thakur, A.; Hota, C. (2014): Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences*, vol. 278, pp. 488-497.

Tomar, K.; Tyagi, S. S. (2014): HTTP packet inspection policy for improvising internal network security. *International Journal of Computer Network and Information Security*, vol. 11, no. 6, pp. 35-42.

Tupakula, U.; Varadharajan, V.; Mishra, P. (2019): Securing SDN controller and switches from attacks. *International Journal of High Performance Computing and Networking*, vol. 14, no. 1, pp. 77-91.

Xu, L.; Huang, J.; Hong, S.; Zhang, J.; Gu, G. (2012): Attacking the brain: races in the SDN control plane. *IBM Research*, vol. 126, pp. 32-36

Yao, J.; Han, Z.; Sohail, M.; Wang, L. (2019): A robust security architecture for SDN-based 5G networks. *Future Internet*, vol. 11, no. 4, pp. 1-14.