International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4
DOI: https://doi.org/10.17762/ijritcc.v11i4.6443
***Article Received**: 02 February 2023 **Revised**: 07 March 2023 **Accepted**: 15 March 2023*

A Secured Software Defined Network Architecture for Mini Net using POX Controller

Vasantharaj Karunakaran[1*], Dr. Angelina Geetha[2]
[1]Research Scholar, Department of CSE,
Hindustan Institute of Technology and Science,
Chennai, Tamil Nadu 603103, India.
vasantharajk35@gmail.com
[2]Professor(CSE) & Dean(ET),
Hindustan Institute of Technology and Science,
Chennai, Tamil Nadu 603103, India.
angelinag@hindustanuniv.ac.in

Abstract: SDN (software-defined networks) is a new technology that stems from numerous network security enhancements. It handles network data in a flexible manner using highly secure frameworks. The secure SDN model's purpose is to ensure data security. The proposed idea to be executed is a robust firewall protection in a mini net employing a POX controller. In order to deal with network-induced dangers, the huge network connectivity influenced environment requires additional protection. The proposed effort focuses on creating a secure SDN simulation architecture that is managed by Open Source POX Controller. Through a POX-controlled traffic management system and a Fingerprint-enabled authentication technique, the system provides multilayer security. The enhanced security is achieved by assessing network traffic as either elephant or mouse flow and selecting the appropriate security level based on data complexity. Mininet is run in a virtual cloud, where protocols and tools are tested and supported by a virtual machine (VM). The novelty is to produce a secure SDN topology was created using a python-based POX controller in the suggested technique. It also provides a low-cost solution as well as rapid development in conjunction with industrial networks.

Keywords*:* Software Defined Networks, firewalls, network security, Mini net, virtual machines.

I. INTRODUCTION

SDN is created without any usage of hardware hence the model is economically suited for all enterprise structure. Firewall is commonly used for securing the incoming network packets communicated from various sources and network nodes [1]. Various topologies are available to manage the SDN enabled network. It incorporates a dynamic adaptive approach to network management and improves the performance of the network through monitoring and controlling operations [2]. SDN controllers are helpful for aligning the network path through continuous monitoring of packets flow. It contains a dedicated emulator and controller that perform the complete topology variations. Mini net is the frequently used open-source window for SDN. Mini net is employed to create a network, support and does research on developed SDN model for improving the performance adequately.

POX is the controller developed by python used in the current scenario [3]. Machine learning approaches are used in the validation phase of the SDN architectures, in which the complete authentication despite security paradigm is done by standard algorithms. Machine learning is incorporated with SDN architectures for many reasons [4].

Software-defined networks have gained Momentum in recent years that resolved challenging problems in the network and follow adaptive security aspects to solve the various security issues. [5] The software-defined network has predominantly become the flexible feature containing network that provides list computation time and optimum meaning at topology. The utilization of available ports in the network significantly increased with the SDN and the architecture used is Mini net [6].

Traffic controlled mini net architecture with the balancing which you LSD and evaluated here with top instrumental setup with load balancing in fat-tree topology with two mini net emulators or develop the year. The system was proposed with a 50% improvement in an average load of 41% improvement in average delay. The considerable improvements response bandwidth and utilization of throughput or increase with the percentage system [7].

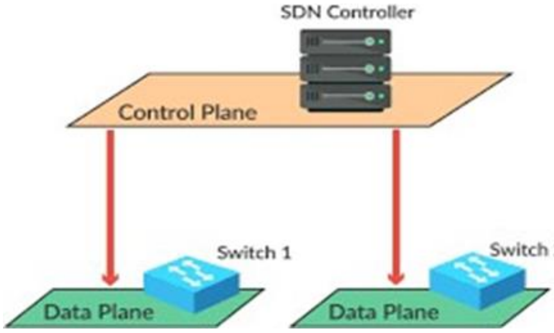

Fig. 1 SDN Open Flow Protocol

Fig. 1 Shows the Open flow protocol model of SDN. The controller mechanism takes over the flow validation process before it reaches the network core. The switches enable the entry-level validation, SDN controller contains the master control over the selection of network path depending on the flow. Elephant flow or Mice flow is commonly determined before the selection of route.

The presented paper is formulated as, Literature study on the existing system in section 2. System tool selection in Section 3. Followed by Section 4. With system architecture. The results obtained from the present implementations are discussed in Section 5. The presented paper is concluded and further, the extension is discussed

II. LITERATURE SURVEY

In [8] A novel SDN-based Gateway is developed for a Healthcare system in a smart environment is presented. The author detects the IoT attacks, tackle the SDN architecture and informed the framework to make the data sharing in a challenging way. The virtual machines are protected by the SDN architecture from external attacks and identify the text with effective validation. Using POX controller and mini net emulator the SDN architecture performs increasingly better compared with different test scenarios.

In [9] the authors proposed a Dynamic packet forwarding and verification scheme in a software-defined network environment is presented here. Like a traditional IP networking SDN network or used for different topology applications full stopper presented system perform the dynamic verification approach and further open-source loading controller is evaluated to test the minute simulations and hard work that is in symbol with the given experimental. 97% accuracy with verification accuracy less than 5% and the throw for degradation less than 10% is achieved.

In [10] the authors proposed a Buffer assistant network-related updates in the time controlled software-defined in-network is developed here. The data plan is still disturbing and manipulating during the fast flow of network sharing. The presented system considers the worst condition of time manipulation and switches the buffers during updates. Percentage system with the set of efficient algorithms to perform the entire problem in polynomial time extensive evaluations in mini net.

In [11] the author presented a ***s***ystem in which SDN enabled quality of service(QoS) development is incorporated in Wide-area measurement systems (WAMS). Through various traffic mechanisms with content-enabled queuing, maximum capacity analysis, critical traffic analysis low latency WAMS are discussed. SDN plays a significant role in organizing the routes for each network.

In [12] the authors discussed about the SDN enabled networks with live streaming of VIDEO in virtual machines are discussed. By evaluating multi-cast streaming video requests, the quality of the service is disturbed. An efficient meta-heuristic algorithm is discussed in which the multi-cast emulators are modelled using SDN controllers. The results perform comparatively better with similar multi-cast networks.

In [20] the authors proposed a secure trusting mechanism (ESTRM) that encourages a node to choose a different path if the current path frequently fails to deliver data to the destination[20]. The system has a limited understanding of how network malware behaves. Defenders must always be able to predict malware behaviour, especially bot behaviour, as well as bot size distribution. As a result, by replaying routing data, the proposed technique protects software-defined networks from serious attacks.

In [21] the authors proposed proposes a technique DPLBAnt which detects the Elephant flow by using a pair of classifiers on both Switches and Controllers. Most Elephant flows will be traced on the Switch which results in accurate and efficient flow detection.

III. SYSTEM DESIGN

The development of SDN infrastructure without organizing the flow variations leadsto resource wastages in the virtual environment. The SDN is a rapidly developed network model that performs better comparing other network architectures [13] [14] whereas utilization of SDN model in proper resource served network is mandatory in terms of economic benefits. The proposed model planned to implement a complete organized [15] SDN architecture for Mini net considers the parameters like security, flow control, network control and optimization in a single platform [16,17-18].

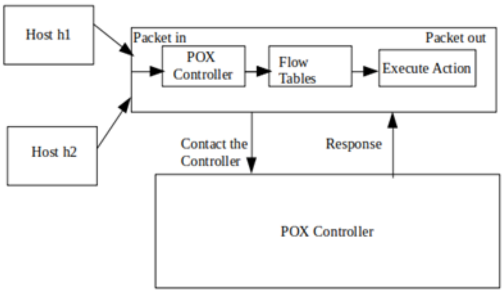

Fig 2.POX Controller Equipped SDN Architecture

Fig 2 presents general SDN system design utilized POX controller developed with Python. [19] Selection of dataset is a complex as well as difficult ones, here a technical dataset that is used on live data retaining value depending on its new recording is considered. The amount of real-time processing that can be done by analyzing the data collected can reduce a lot of manual work and enable patterns in traffic data that occur over a large period that a human cannot identify. These logs also allow us to see the amount of data being transferred and allow organizations to allocate bandwidth depending based on the future scope of usage patterns. Machine learning models [17] are incorporated to understand, interpret and apply the pattern of traffic flow for further research.

IV. SYSTEM ARCHITECTURE

The system architecture in Fig. 3 consists of switches connected with the network followed by a POX controller that detects the traffic range in the magnet. The traffic is classified into elephant flow or mice flow to stop the Mini net has the topology that is created with the dedicated topology controller and hence the complete SDN execution is completed.

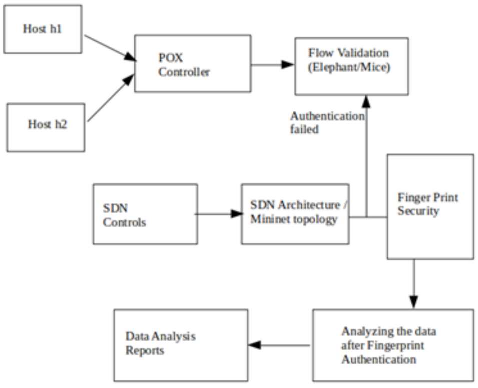
Host h1
Host h2
POX Controller
Flow Validation (Elephant/Mice)
Authentication failed
SDN Controls
SDN Architecture / Mininet topology
Finger Print Security
Data Analysis Reports
Analyzing the data after Fingerprint Authentication

Fig 3 Improved SDN Architecture using POX

The network device connected with the functional operation of layer 2 of the OSI model is called the data link layer. The data link layer receives the data delivered by each device connected to the physical ports of the network and transmit the data again back through the ports given by the devices. The required packets are transmitted between the ports. When the device is linked to your switch the control records the media Axis address encode the device information in the network interface with the switch via Ethernet cord to stop the Mac address is being used to identify the device connected with the outgoing packets and incoming

POX is open-source connectivity of SDN controller developed by Python. POX controller is developed and tested the network applications rapidly for stop the mini net maids eventual mission including the POX controller will stop the user connected via PO controller to turn flow table devices into hubs, switches, load balancers and firewalls etc. By this process, the traffic flow is detected and the state control is forwarded to the next flow process. Elephant flow is created as a flow process that is obtained in collisions network congestion and transmission delay since a large span of packets or congested to the node links. Simultaneously the most flow has been able to gain adequate bandwidth causing the transmission delay to grow.

Mini net is a platform that allows network tools and protocols to be incorporated with the given topology for testing and development. Mini net can be constructed in a realistic way inwhich it incorporates the network that contains different types of system operation and services. Examples: Virtual Machines, cloud-hosted and native. Mini net offers low-cost solutions as well as expertise development synchronized with the production networks. Many attacks of emerging network protocol [14] the taxes a prototype used for verification of the complicated topologies without the requirement of expensive gear changes. It performs real time code on UNIX and LINUX kernels to provide realistic executions. Huge community contribution or required to access the open-source environment.

4.1 Implementation Summary

The overall performance of the SDN enabled Mini net is based on a various selection of parameters and blocks like, Flow controller using POX, Flow range estimator, SDN emulator, Fingerprint authentication etc. The novel methodology is presented here with the SDN controlled Mini net is presented. The POX controller completely takes over the architecture major blocks and their operations. The SDN enabled mini net protocol completely isolated andestablish the connection on demand. The packet flow is initiated once the network is established. The flow of the data packets is validated first with the elephant or Mice flow monitor.

DOI: https://doi.org/10.17762/ijritcc.v11i4.6443
***Article Received**: 02 February 2023 **Revised**: 07 March 2023 **Accepted**: 15 March 2023*

Network switches initially take over the entry of data packets via the demanded network. The fingerprint enabled system authenticates the entry of user requests in the mediate and formulates the further route. The overall step taken by the network structure is recording the log files. Based on the recorded log entry further machine learning algorithms are utilized in future developments. The detection of anomaly entry is feasible with the present structure further incorporation of machine learning algorithms.

V. Results and Discussions

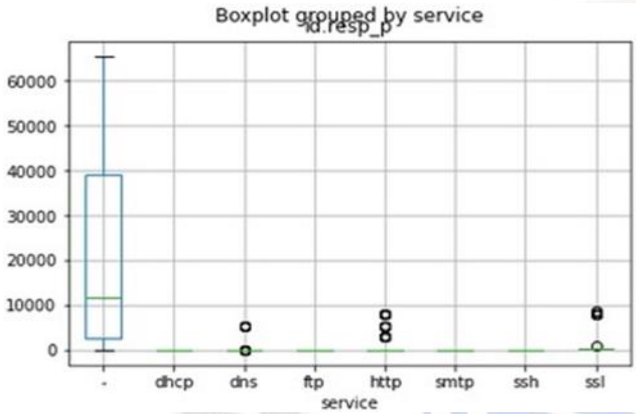

Fig. 4 Service via Data packets transferred

Fig. 4 Shows the service concerning equivalent data flow is plotted through Box plot.From the presented SDN model, the normalization of flow is identified.

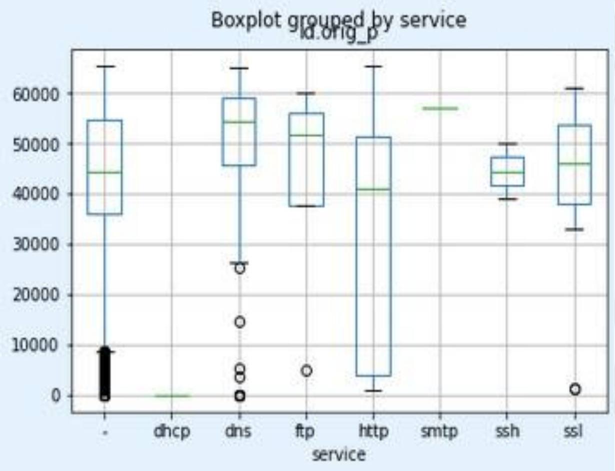

Fig. 5 Organized Packet flow via SDN mini net

Fig. 6 represents the services that show a huge correlation between the ORIG and RESP bytes as well as Port numbers. To summarize this large correlation of features and services, we are using seaborn correlation matrix

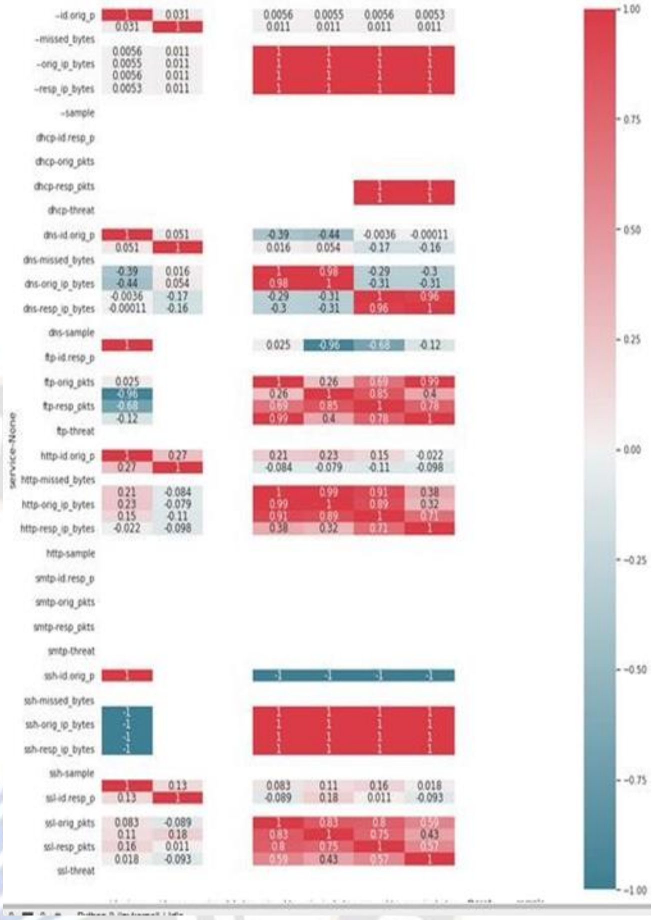

Fig. 6 Visualization of a Correlation matrix

From the visualization of Fig. 5, we can see the high correlation between all the fields with numbersclose to 1. It summarizes the behavior of the traffic that was recorded in our dataset. This plot reduces the number of plots/figures that show the relationship between the features.

5.1 Experimental Analysis

Extra Feature Information have access to other log files, also generated by BRO, we can triangulate the correlation in attacks from all of the files in realtime. The log files can be: **httplog** (contains features representing HTTP traffic that represents the majority of our traffic data)

host_detections log (HOST - PORT pair data that gave us multiple probabilities of attacks can be confirmed using the features from the host_detections log file)

malicious domains log (this file contains a list and behaviour of all the malicious domains relating to which we can confirm the attacks)

originating IP bytes log (recorded by BRO at the same time and has additional IP features)

**honey pot log ** (most common exploits and honeypots database that records their behavior)

Here we will use column indexing and compare the port number with the service to indicateif there is an attack or a potential threat. Using Port numbers and service is a very potent way of identifying potential threats. We know the standard port number for services. For example, http uses 80, SSL uses 443, and so on. We will use this to our advantage to scan for abnormal port usage patterns.

5.2 Performance Metrics

For our purposes, the following performance measures are used to evaluate the correlation matrix. The following are the formulas for calculating the performance parameters:

- Accuracy : (TP + TN)/TOTAL
- Misclassification Rate : (FP+FN)/TOTAL
- True Positive Rate : TP/Actual Yes
- False Positive Rate : FP/Actual No
- Specificity :TP/Actual No
- Precision :TP/Predicted Yes
- Prevalence : Actual Yes/Total
- Recall : TP/(TP+FN)
- F1 Score :2* ((Recall*Precision)/(Recall+Precision))

VI. CONCLUSION

We discretize the continuous-time into a slot for the sake of tractability in packet delivery and other relevant routing operations. The most important assumption in this work is that fingerprint authentication is done using a POX controlled mini net. The continuous learning and transfer learning model of the SDN network is evaluated here. The presented system focused on organized SDN architecture that works out well in all the blocks of the network. It contains the flow control mechanism through POX controller with elephant or Mice flow. Based on the analysis in each blocks the SDN network override the performance and route the data packets efficiently. The significant results are achieved via the proposed Fingerprint mechanism using a machine learning-based pattern verification system. Further, the system needs to be improved by utilizing an ensemble machine learning process for multi-level validation and providing robust security.

Declaration of Conflicts of Interest:

The authors declare that they have no conflict of interest.

Author Contributions:

Corresponding author contributed in the Abstract, Introduction, System Design and results, while the co-author contributed in designing and framing the structure of the paper

REFERENCES

[1] Kaur, S., Singh, J. and Ghumman, N.S., August. Network programmability using POX controller. In *ICCCS International conference on communication, computing & systems, IEEE* (2014), (Vol. 138, p. 70). sn.

[2] Srinivasa, K.G. and GM, S., Introduction to Data Analytics. In *Network Data Analytics* 2018, (pp. 3-28). Springer, Cham.

[3] Miner, G., Elder, J., IV, & Hill, T,. Practical text mining and statistical analysis for non-structured text data applications. (2012), Cambridge: Academic Press.

[4] Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F. and Caicedo, O.M., A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications, 9(1), (2012), pp.1-99.

[5] B. Ahmed, N. Ahmed, A. W. Malik, M. Jafri and T. Hafeez, "Fingerprinting SDN Policy Parameters: An Empirical Study," in IEEE Access, vol. 8, pp. 142379-142392, 2020, doi: 10.1109/ACCESS.2020.3012176.

[6] M. W. Hussain, K. H. K. Reddy, J. J. P. C. Rodrigues and D. S. Roy, "An Indirect Controller- Legacy Switch Forwarding Scheme for Link Discovery in Hybrid SDN," in IEEE Systems Journal, vol. 15, no. 2, pp. 3142-3149, June 2021, doi: 10.1109/JSYST.2020.3011902.

[7] S. Ejaz, Z. Iqbal, P. Azmat Shah, B. H. Bukhari, A. Ali and F. Aadil, "Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function," in IEEE Access, vol. 7, pp. 46646-46658, 2019, doi: 10.1109/ACCESS.2019.2909356.

[8] Y. Meng, Z. Huang, G. Shen and C. Ke, "SDN-Based Security Enforcement Framework for Data Sharing Systems of Smart Healthcare," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 308- 318, March 2020, doi: 10.1109/TNSM.2019.2941214.

[9] Q. Li, X. Zou, Q. Huang, J. Zheng and P. P. C. Lee, "Dynamic Packet Forwarding Verification in SDN," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 6, pp. 915-929, 1 Nov.-Dec. 2019, doi: 10.1109/TDSC.2018.2810880.

[10] X. He, J. Zheng, H. Dai, Y. Sun, W. Dou and G. Chen, "Buffer-Assisted Network Updates inTimed SDN," in IEEE

Transactions on Communications, vol. 69, no. 10, pp. 6822-6837, Oct. 2021, doi: 10.1109/TCOMM.2021.3093929.

[11] M. Rezaee and M. H. Yaghmaee Moghaddam, "SDN-Based Quality of Service Networkingfor Wide Area Measurement System," in IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3018-3028, May 2020, doi: 10.1109/TII.2019.2893865.

[12] A. Erfanian, F. Tashtarian, A. Zabrovskiy, C. Timmerer and H. Hellwagner, "OSCAR: On Optimizing Resource Utilization in Live Video Streaming," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 552-569, March 2021, doi: 10.1109/TNSM.2021.3051950.

[13] Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K. and Turletti, T., 2014. “A survey of software-defined networking: Past, present, and future of programmable networks”. IEEE Communications surveys & tutorials, 16(3), pp.1617-1634.

[14] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, “Review: A survey on routing techniques in underwater wireless sensor networks,” *J. Netw. Comput. Appl.*, vol. 34, no. 6, pp. 1908–1927, Nov. 2011.

[15] N. Li, J.-F. Martínez, J. M. Chaus, and M. Eckert, “A survey on underwater acoustic sensornetwork routing protocols,” *Sensors*, vol. 16, no. 3, p. 414, Mar. 2016.

[16] P. Xie, J.-H. Cui, and L. Lao, “VBF: Vector-based forwarding protocol for underwater sensor networks,” in *Proc. IFIP-TC6 Netw.* Berlin, Germany: Springer-Verlag, 2006, pp. 1216–1221.

[17] Klaine PV, Imran MA, Onireti O, Souza RD. “A survey of machine learning techniques applied to self organizing cellular networks”. IEEE CommunSurv Tutor. 2017; PP(99):1.

[18] Nunes, B.; Mendonca, M.; Nguyen, X.; Obraczka, K.; Turletti, T., "A Survey of Software- Defined Networking: Past, Present, and Future of Programmable Networks," Communications Surveys & Tutorials, IEEE ,vol.PP, no.99, pp.1,18

[19] Benson T. Data Set for IMC 2010 Data Center Measurement. 2010. http://pages.cs.wisc.edu/tbenson/IMC10_Data.html. Accessed 28 Dec 2017.

[20] Abdulrahman Saad Alqahtani, “Security threats and countermeasures in software defined network using efficient and secure trusted routing mechanism”, Computer Communications, Volume 153, 2020, Pages 336-341, ISSN 0140-3664,.

[21] Mosab Hamdan, Suleman Khan, Ahmed Abdelaziz, Shahidatul Sadiah, Nasir Shaikh-Husin, Sattam Al Otaibi, Carsten Maple, M.N. Marsono, DPLBAnt: Improved load balancing technique based on detection and rerouting of elephant flows in software-defined networks, Computer Communications, Volume 180, 2021, Pages 315-327, ISSN 0140-3664