International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST22913

Recent Advancements in Intrusion Detection in Software Defined Network Security

Shridhar R Sharma*[1], V. Mohan[2], B K Madhavi[3], S B Kishor[4]

*[1]Associate Professor, Department of Electronics , J M Patel College, Bhandara, , Maharashtra, India

[2] Research Scholar, CHLR (Computer Science), SP College, Chandrapur, Maharashtra, India

[3] HOD, Department of Computer Science and Engineering Department, NMR Engineering College, Hyderabad, India

[4] HOD, Department of Computer Science, SP College, Chandrapur, Maharashtra, India

Article Info
Volume 9, Issue 1
Page Number : 35-42

Publication Issue
January-February-2022

Article History
Accepted : 01 Jan 2022
Published : 04 Jan 2022

ABSTRACT

In recent years, Software Defined Networking (SDN) has enabled total control over the network's data flow. SDN acts as a centralised point of administration for data and traffic management. Due to the fact that SDN is an open source software, it is more vulnerable to security concerns. Security policies must also be adhered to, since this would expose the controller to the greatest attacks. DDOS and DOS assaults are more prevalent in SDN controllers. DDOS is a damaging assault that disrupts the usual flow of communication and initiates an overflow of flooded packets, thereby shutting down the system. Machine Learning approaches assist in identifying the network's hidden and unexpected patterns, hence aiding in the analysis of the network's flow. All classified and unclassified approaches can assist in detecting hostile flows depending on specific factors such as packet flow, time length, precision, and accuracy rate. To identify DDOS assaults, researchers employed Bayesian Networks, Wavelets, Support Vector Machines, and KNN. According to the review, KNN offers superior results due to its increased accuracy and reduced false positive rate for detection. We explore the various strategies used in DDoS detection and examine new improvements in intrusion detection in software defined networks in this article.

Keywords : Software Defined Networking, Bayesian Networks, Wavelets, Support Vector Machines, DDoS

I. INTRODUCTION

Intrusion detection and prevention systems (IDPS) are critical, particularly for enterprise networks. They enable the defence of the internal network by monitoring network traffic, alerting the administrator when suspicious traffic is detected, and filtering or redirecting suspicious traffic as necessary. Fig. 1

depicts a straightforward network-based IDPS system. A network-based IDPS's typical components are an IDPS sensor, an

IDPS management server, and an IDPS console [1]. The IDPS sensor is in charge of monitoring and analysing network traffic. The management server is a centralised device that collects and handles data from sensors. A console is an application that acts as an interface for users and administrators of the IDPS. Additionally, the boundary switch must replicate each packet to the IDPS sensor. This may be accomplished by setting the standard switch's spanning port. This port has visibility into all network traffic that passes through the switch. The introduction of Software-Defined Networks (SDN)

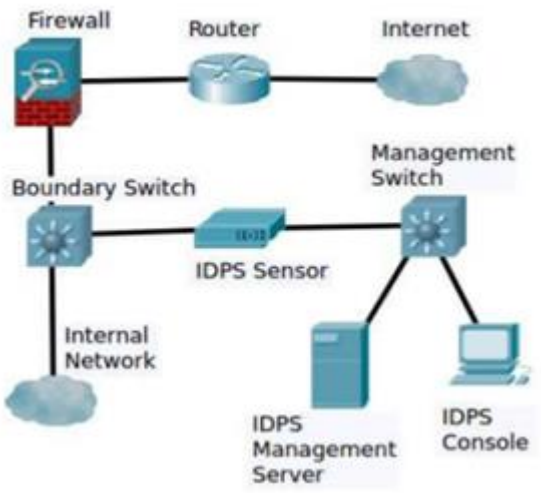

Figure 1: A Simple Network-Based IDPS System

[2] enables the implementation of intrusion detection systems.

SDN has garnered considerable research attention in recent years because to its capacity to solve the lack of programmability in traditional network architectures and to allow easier and quicker network innovation. SDN decouples the data plane from the control plane and enables networking applications to be implemented on top. As seen in Figure 2, the SDN architecture is composed of three levels.

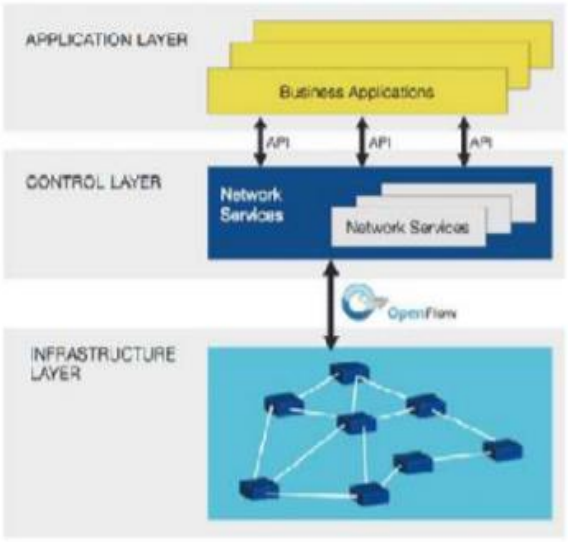

Figure 2: Architecture of a SDN

The infrastructure layer, also known as the data plane, is the most fundamental layer, and it is comprised of the forwarding network elements. Its primary functions include data transmission, local information monitoring, and the collection of statistical data. The control layer, also known as the control plane, is located in the centre of the hierarchy. It is in charge of programming and overseeing the operation of the forwarding plane. This layer, which is located on top of the architecture, comprises network-oriented applications.

The controller provides an abstracted and global view of the forwarding plane to the application layer, which may be used to make decisions. It then makes use of this information to offer suitable instructions to the control layer or to operate the network services.

Follows: Section II describes the related work, Section III describes about existing in intrusion detection methods. Section IV talks about recent advancements in intrusion detection methods. Section V gives the conclusion.

II. SOFTWARE-DEFINED NETWORKS AND RELATED WORK

The networking paradigm of Software-Defined Networks (SDN) separates the control and forward planes 2]. The devices provide data flow switching capabilities, while the control plane is separated to create a new entity called the network controller. At the bottom of the stack is the forward plane, which contains hardware devices such as switches, routers, firewalls, and intrusion detection systems (IDS). The devices lack the necessary software intelligence to populate the forwarding tables. The network logic was transferred separately to the controller layer. The controller abstracts the devices and offers the resources necessary for low-level forwarding device programming.The controller, also known as the Network Operating System (NOS), delivers network state and topology information. Additionally, the controller communicates with the apps via northern and southbound APIs. The southbound API, on the other hand, is used to facilitate communication between the controller and forwarding devices. OpenFlow serves as the de facto southbound protocol for SDN 3]. On top of the SDN model stack is the application plane. Network programmability is a critical capability of the SDN paradigm, since it enables applications on the top plane to access physical devices via the controller. Programmability enables and accelerates innovation across a vast array of network applications, including monitoring, traffic engineering, security, and cloud computing. The SDN architecture is fundamentally centralised. A controller is a centralised entity that gives a comprehensive view of the whole network; it simplifies the process of managing and enforcing regulations. Additionally, it reduces the number of errors associated with creating and installing network policies. Centralization improves network resilience and interoperability; for example, many devices from disparate industrial sectors may be merged and abstracted into a single network. In typical networking systems, security risks are a significant problem. The attacks are becoming more sophisticated in SDN networks. Numerous advantages of the paradigm are coupled with new risks that were not achievable in older networks. For the southbound Open Flow protocol, a security investigation revealed a variety of threats developed from the SDN standard protocol, such as denial of service attacks on flow tables and on the devices' control channels between the devices and controller (DoS). Conflicts between application privileges cascade to flow rules. The control channel between the controller and the switch is established using TCP, with the option of encrypting the channel using the Transport Layer Security (TLS) protocol. Without encryption, communication between the controller and forwarding devices is vulnerable to man-in-the-middle attacks. Kloti et al. examined the OF protocol's security 4]. According to the study, denial of service attacks have posed a danger to the flow tables and communication channels by flooding those components with OpenFlow rules and requests. Additionally, by installing rules from untrusted sources, tampering attacks have mostly targeted the flow tables on the devices. Kreutz et al. found that there are seven potential dangers to SDN 5]. Three dangers are inextricably tied to the controller:

- Attacks on the controller's communications with the data plane devices.
- Vulnerabilities in the controller
- Untrusted programmes were used to launch attacks on the controller

Intrusion Detection Systems (IDS) are software or hardware-based systems that monitor network traffic for potential security threats. The standard intrusion detection technique consists of three phases: gathering network data, processing it, and finally initiating a correct reaction if a danger is discovered. There are three methods for analysing collected traffic: signature-based analysis, anomaly detection

analysis, and specification-based analysis. To begin, signature-based, in which a system maintains a database of predetermined violation signatures and compares them to network activity signatures. Secondly, anomaly or outlier analysis is concerned with the system's ability to distinguish between normal and aberrant patterns. Normal activities are indicated for the system in a baseline profile that the system produces throughout the learning phase. Thirdly, stateful protocol analysis; in this technique, a preset pattern of protocol behaviour is constructed, network activity is compared to the expected behaviour described by protocols, and an alarm is raised in the event of profile violation. To optimise IDP performance, a mix of strategies is applied 6]. The signature-based technique has a key limitation in that it cannot identify new threats, whereas anomaly detection has a greater risk of false alarms. A mixed approach is used in the majority of commercial deployments 7]. Anomalies, sometimes referred to as outliers, are patterns that are unexpected. We presume that invasive or malicious activity is infrequent in the context of networking 8].

III. III. EXIXTING INTRUSION DETECTION METHODS

Intrusion Detection Systems (IDS) continuously monitor the network for odd behaviour in order to detect whether or not it has been hacked. There is a difference between a host-based and a network-based identification system. A host-based intrusion detection system is a software application that operates on a single computer and analyses its own traffic for signs of assault. A network-based intrusion detection system is installed on a separate workstation that monitors the whole network's activities 3]. Intrusions can be detected in two ways: by detecting misuse and by detecting anomalous intrusions. Intrusion detection by misuse, or knowledge-based intrusion detection, is the most prevalent strategy, which compares network traffic to a database of known assaults 4]. When an event fits the signature of an attack stored in the database, an alarm is triggered. Anomaly-based intrusion detection examines network traffic for any divergence from the system's usual or anticipated behaviour 3]. It then uses that knowledge to learn and adapt. Among the benefits of misuse intrusion detection is its precision and low false positive rate. When an intrusion detection system misinterprets regular communication as an attack, this is referred to as a false positive. The drawback of misuse intrusion detection is the ongoing maintenance required to keep the database current. The high rate of false positives is the system's primary downside. This is due to its capacity for change and adaptation throughout time. However, one of its advantages is its capacity to "identify efforts to exploit novel and unknown weaknesses," potentially resulting in the discovery of new attacks 4]. Numerous strategies are used to process the data that arrives the Intrusion Detection System. The following sections describe many of the most frequently used methodologies, including the neural network methodology employed in this study:

i.Expert Systems - data is compared against an audit trail based on a previously determined set of attack rules. Similar to expert systems, signature analysis converts the "semantic description of an assault into the proper audit trail format" 5]. This is a technique that is frequently employed in commercial systems such as Stalker, Real Security, and Cisco IDS.

ii.ColoredPetrinets -Using expert knowledge bases, Colored Petri Nets provides graphical representations of assaults. Statistical Analysis - the data's behaviour through time is compared to a variety of factors. These variables include, but are not limited to, user logon, storage space, RAM, and CPU utilisation.

iii.Data Mining - excels at obtaining "unknown but possibly relevant data from massive data repositories" 5]. Neural Networks — a learning method is used to compare the input and output vectors. For intrusion detection, multiple neural network-based techniques were used. Jake Ryan, Meng-Jang Lin, and

RistoMiikkulaine of The University of Texas at Austin demonstrated that their approach was 96 percent accurate in detecting unexpected network activity12]. Another experiment led by Griffin University's Robert Birkely achieved a classification rate of 100% for normal, 92% for known attacks, and 80% for unknown assaults 13].

Additionally, Kohonen's Self-Organizing Map was employed to identify intrusions. In contrast to backpropagation, self organising maps (SOM) are unsupervised learning networks with unknown outputs. Experiments with SOMs have also demonstrated a high degree of effectiveness in classifying regular traffic from harmful traffic. Within a distance of zero to three, Brandon Rhodes, James Mahaffey, and James Cannady were able to classify all regular traffic. Whereas assault sessions were categorised at a distance of eighty to six hundred thirty - suggesting a significant deviation from the usual 14].

IV. RECENT ADVANCEMENTS

A FLEXIBLE NETWORK-BASED INTRUSION DETECTION SYSTEM

With the ability of centralization in SDN environment, the problem of IDS will be easily overcome. The general architecture is given in Fig 4.

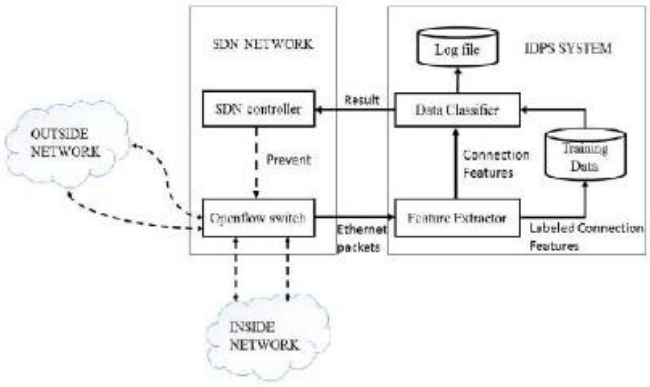

Figure 3 : IDS based on SDN Architecture

i.SDN controller: An OpenFlow controller is a software-defined networking (SDN) programme that controls flow control in an SDN context. The OpenFlow protocol is used to control SDN controllers. The SDN controller serves as the network's operating system (OS). The controller is the point of contact for all communications between applications and devices. The Open-Flow protocol establishes a connection between controller software and network devices, allowing server software to instruct switches on where to transmit packets. The controller configures network devices and determines the optimal routing for application traffic using the OpenFlow protocol.

ii. OpenFlow switch: OpenFlow is a protocol that enables the programming of the flow table in various switches and routers. Three components comprise an OpenFlow switch: a Flow Table, a Secure Channel, and the OpenFlow Protocol. One of the advantages of Openflow switches is their capacity to duplicate packets for the purpose of inspection. Additionally, the Openflow option functions as a firewall, preventing several assaults.

iii. Feature Extractor: Choosing an appropriate collection of features is difficult owing to the diversity of protocols present at the network layer and the number of features collected from each protocol. The current state of network intrusion detection research does not provide a comprehensive feature set capable of detecting all network-level intrusions 8], and our article focuses on a specific class of intrusions in terms of detection scope, DOS, and Probe. As a result, some elements are taken from the header dependent on the TCP connection. Additionally, depending on the mode selected, all extracted features are transmitted to the training module; alternatively, features are utilised to check for normal or bad connections.

iv. Data Classifier: Using machine learning, a classifier determines if this is an assault or not. Numerous models, such as decision trees and support vector machines, can be implemented. The classifier is trained using TCP and UDP flooding and then put to

the test using a real-world DDoS flooding assault. If the classifier detects any malicious traffic, the IDS instantly alerts the SDN controller.

v. Log Module: This module is capable of logging assaults and generating data for the training process.

B. BLOCK CHAIN BASED INTRUSION DETECTION

i. SDN Controller on Blockchain

Specifically, the research of blockchain-based distributed SDN control plane for secure information updating between controllers is connected to our work. For example, the proposal 15uses a permissioned blockchain to keep track of system updates and time stamps in each controller. Despite using SPBFT consensus to broadcast messages/requests, the work lacks a consensus process to add new participants and ignores controller insider assaults. Our technique, however, addresses these.

ii. Multiple SDN Controllers

Using several controllers with collaborative detection communication, we attempt to decentralise the SDN control plane in this study. An AS managing numerous controllers that straddle geographical domains, or a single domain with multiple controllers/firewalls providing various networking inbound points. Using this criteria, we assume no external attacks are threatening controller coordination communication, and we focus on insider attacks that compromise controller nodes and exploit communications.

iii. Risk Model

We can now develop the danger model that encourages us to guard against, based on our strategy employing blockchain-based controllers P2P network with collaborative detection. Decentralized and coordinated controllers are studied and addressed to solve the first challenge, which is against the availability of traditional centralized intelligence utilising a single SDN controller. An (inside) attacker can tamper with the sent information if the collaborative defensive communication between controllers is not protected. In the joint defensive communication, we use blockchain to protect the transactions (carrying the detection information). We think no outside attacker may get such inside defence information because we assume our collaborative defence communication will not cross the network barrier. There is no confidentiality issue because only insiders should have access to the material, however we must register all insiders/participants. So we utilise permissioned block chain to regulate who may participate in the coordinated defence. Inside attackers that utilise real identities or enrolled as lawful users at the participant registration stage are considered compromised controllers. To overcome this issue, we propose the PBFT consensus protocol, where every transaction carrying defensive information (i.e. detection signature) must be certified by a quorum of other participants.

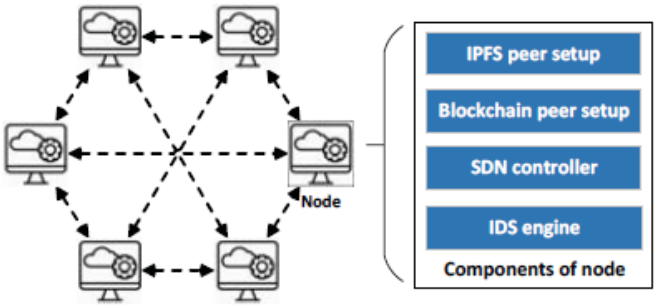

Figure 4 : Block chain-enabled SDN with four steps: Generate, Import, Fetch and Update

C. HYBRID APPROACH

According to the current discoveries, the architecture for identifying anomalous flows in an SDN context is based on the notion of transductive Confidence Machines. It is calculated using the detection point's confidence level and the probability p to determine whether the detection point can be recognised or not. To determine the detection point based on the p value, it is discovered that the higher the p value, the more probable the detection point exists on the controller.

This technique detects assaults by utilising a KNN algorithm strategy that takes strangeness and independence into account as parameters.

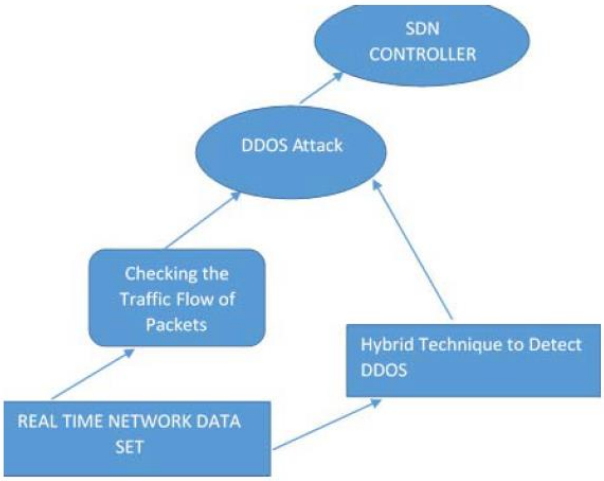

Figure 5 : Hybrid IDS

i) Because the patterns of abnormal behaviour in traffic are so fast, a hybrid method enables detection and mitigation to occur more quickly and easily.

ii) Because the KNN method computes the Euclidian distance and sorts the data sets based on that value, the detection rate is quite sluggish. After comprehending the outcome, a comparison of the result utilising an optimization strategy is performed.

iii) Using noisy data and the duration of the packets' flow, the combined technique of SVM and ANN lengthens the testing period, leading in a greater accuracy rate and detection crucial.

iv) By combining SVM and ANN approaches, the difficulty of finding an ideal technique for detection based on noisy data is increased as well.

v) The experiments are conducted using the KDDCUP99 Datasets, which are also used to develop the algorithms. The data sets are a continuous real-time stream of data that significantly improves the realism of intrusion detection. This data collection contributes to the algorithm's increased detection precision.

V. CONCLUSION

SDN (software-defined networking) has become the industry standard for network management because it isolates the flow control logic from the data layer. Specifically, in this paper, we investigated a block chain-enabled collaborative intrusion detection system in SDN networks, which provides trust management of the controllers as well as integrity of the detection signature sharing over the controllers in order to gain a coordinated defence and defend against insider attacks that are supported by n-compromise immunity. It was also examined in this research, as well as its limits and the advantages of utilising a hybrid strategy, which provides a greater accuracy rate while maintaining a lower precision value.

VI. REFEERNCES

[1]. Y. Jarraya, T. Madi, and M. Debbabi, “A survey and a layered taxonomy of software-defined networking,” IEEE communications surveys & tutorials, vol. 16, no. 4, pp. 1955–1980,2014.

[2]. K. S. Sahoo, S. Mohanty, M. Tiwary, B. K. Mishra, and B. Sahoo, “A comprehensive tutorial on software defined network: The driving force for the future internet technology,”in Proceedings of the International Conference on Advances in Information CommunicationTechnology&Computing.ACM,2016,p.114.

[3]. P.Maiti,J.Shukla,B.Sahoo,and A. K.Turuk,“Qos-awarefognodesplacement,”in20184thInternational Conference on Recent Advances in Information Technology (RAIT). IEEE, 2018, pp.1–6.

[4]. “Mathematical modeling of qos-aware fog computing architecture for iot services,”in Emerging Technologies in Data Mining and Information Security. Springer, 2019, pp.13–21.

[5]. M. Tiwary, D. Puthal, K. S. Sahoo, B. Sahoo, and L. T. Yang, “Response time optimization for cloudlets in mobile edge computing,” Journal ofParallel and Distributed Computing, vol. 119, pp. 81–91,2018.

[6]. K. S. Sahoo, D. Puthal, M. Tiwary, J. J. Rodrigues, B. Sahoo, and R. Dash, “An early detection of low rate ddos attack to sdn based data center networks

using information distance metrics," Future Generation Computer Systems, vol. 89, pp. 685–697,2018.

[7]. K. Sahoo, K. S. Sahoo, and M. Tiwary, "Signature based malware detection for unstructured data in hadoop," in Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on. IEEE, 2014, pp.1–6.

[8]. Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir, "Application-awareness in sdn," in ACM SIGCOMM computer communication review, vol. 43, no. 4. ACM, 2013, pp.487–488.

[9]. R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in Local Computer Networks (LCN),2010 IEEE 35th Conference on. IEEE, 2010, pp.408–415.

[10]. Y. Zhang, "An adaptive flow counting method for anomaly detection in sdn," in Proceedings of the ninth ACM conference on Emergingnetworking experiments and technologies. ACM, 2013, pp.25–30.

[11]. S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, "A survey of securing networks using software defined networking." IEEE Trans. Reliability, vol. 64, no. 3, pp. 1086–1097,2015.

[12]. K.Giotis, C. Argyropoulos, G. And roulidakis, D. Kalo geras and V.Maglaris,"Combining open flow and flow for an effective and scalableanomaly detection and mitigation mechanism on sdn environments," Computer Networks, vol. 62, pp. 122–136,2014.

[13]. R. Kokila, S. T. Selvi, and K. Govinda rajan, "Ddos detection and analysis in sdn-based environment using support vector machine classifier,"in Advanced Computing (ICoAC), 2014 Sixth International Conference on. IEEE, 2014, pp.205–210.

[14]. R. Miao, M. Yu, and N. Jain, "Nimbus: cloud-scale attack detection and mitigation," in Acmsigcomm computer communication review, vol. 44, no. 4. ACM, 2014, pp.121–122.

[15]. Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, and X. Zheng, "Sd-anti ddos: Fast and efficient ddos defense in software-defined networks," Journal of Network and Computer Applications, vol. 68, pp. 65–79,2016.

[16]. J. Ashraf and S. Latif, "Handling intrusion and ddos attacks in software defined networks using machine learning techniques," in Software Engineering Conference (NSEC), 2014 National. IEEE, 2014, pp.55–60.

[17]. K. S. Sahoo, M. Tiwary, and B. Sahoo, "Detection of highrated dos attack from flash events using information metrics in software defined networks," in Communication Systems & Networks (COMSNETS), 2018 10th International Conference on. IEEE, 2018, pp. 421–424.

[18]. S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. Murthy, "A fast iterative nearest point algorithm for support vector machine classifier design," IEEE transactions on neural networks, vol. 11, no. 1, pp. 124–136,2000.

[19]. L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 532, 2001.

[20]. M. Alka sassbeh, G. Al-Naymat, A. Hassanat, and M. Almseid in, "Detecting distributed denial of service attacks using data mining techniques," International Journal of Advanced Computer Science and Applications, vol. 7, no. 1, 2016.

Cite this article as :

Shridhar R Sharma, V. Mohan, B K Madhavi, S B Kishor, "Recent Advancements in Intrusion Detection in Software Defined Network Security", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 9 Issue 1, pp. 35-42, January-February 2022. Available at doi : https://doi.org/10.32628/IJSRST22913
Journal URL : https://ijsrst.com/IJSRST22913