Journal of Engineering Sciences

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

IDS in SDN using RNN

"KRISHNAN K, NITHIN RAJ M, RANJITH KUMAR S,SUGNYA B"

KRISHNAN K Department of Computer Science and Engineering ,*SREC*,India .

NITHIN RAJ M Department of Computer Science and Engineering ,*SREC*,India.

RANJITH KUMAR S Department of Computer Science and Engineering ,*SREC*,India.

SUGANYA B ,Department of Computer Science and Engineering ,*SREC*,India.

ABSTRACT

Software Defined Networking Technology (SDN) provides a prospect to effectively detect and monitor network security problems ascribing to the emergence of the programmable features. Recently, Machine Learning (ML) approaches and Convolutional Neural Network (CNN) have been implemented in the SDN-based Intrusion Detection Systems (IDS) to protect computer networks and to overcome network security issues. A stream of advanced machine learning approaches –the deep learning technology and Recurrent Neural Network (RNN) commences to emerge in the SDN context. In related work, we reviewed various recent works on machine learning, convolutional neural network and deep neural network methods that leverage SDN to implement IDS. More specifically, we evaluated the techniques of deep learning in developing SDN-based IDS. In the meantime, in this proposed model, we covered tools that can be used to develop IDS models in an SDN environment. This model is concluded with a discussion of ongoing challenges in implementing IDS using ML/DL and future works.

Keywords: RNN,attacks.KDD dataset,SDN,CNN.

CHAPTER 1

INTRODUCTION

1.1 Software Defined Network (SDN)

Software Defined Network allows network administrators to customize a given network according to changing user and business needs. Software Defined Network is an emerging architecture that is dynamic, manageable and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's application. Open Networking Foundation (ONF) defines the SDN as ''the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices'' . Basically, the SDN decouples the Control plane (CP) and the Data plane (DP) which allows flexibility and centralized flow control of the entire network as well as provides capabilities of responding rapidly to changing network conditions, business, market and end-user's needs. Network intelligence is centralized in an SDN controller that maintains a global view of the network, which appears to applications and policy engines as a single, logical switch. The goal of Software Defined Network is to improve network control by enabling enterprises and service providers to respond quickly to changing business requirements. SDN architecture comprises of three layers or planes i.e. Infrastructure layer, Control layer, and Application layer. Infrastructure layer/plane: Data/Infrastructure plane consist of the network elements such as physical switches and virtual switches which expose their capabilities towards the control plane. Control layer/plane: It represents the SDN controller software that acts the brain of Software Defined Network. It supervises the network forwarding behavior through an open interface. Application layer/plane: It mainly consists of the end-user business application or functions organisations use, which can include Intrusion Detection System, load balancing or firewalls. These applications can be network management or business applications used to run large data centers. These three layers communicate using Northbound and Southbound APIs. A northbound interface is defined as the connection between application plane and control plane whereas southbound interface is the connection between control plane and the data plane.

1.2 Attacks in SDN

These are the possible attacks in SDN

DoS attack

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users by sending too much traffic to the server. E.g.- ICMP flood, SYN flood, Buffer overflow etc.

Probe attack

Probe attack is an attack in which the attacker scans a machine or a networking device in order to

determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. E.g.- ipsweep, ports weep, Nmap etc.

U2R

User to Root (U2R) attacks are the attacks in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges. E.g.- sniffing, social engineering attack etc.

R2L

Remote to Local (R2L) attack involves unauthorized access from a remote machine. E.g.- imap, ftp write, phf etc.

Network Manipulation

A critical attack that occurs on the control plane. An attacker compromises the SDN controller, produces false network data and initiates other attacks on the entire network.

Traffic diversion

This attack occurs to the network elements at the data plane. The attack comprises a network element to redirect traffic flows and allow eavesdropping.

Side channel attack

The network elements at the data plane can be the target of this attack. Timing information, such as how long a new network connection takes to establish, can inform an attacker if a flow rule exists or not.

App manipulation

This attack takes place in the application plane. An exploit of application vulnerability could cause malfunction, disruption of service, or eavesdrop of data. An attacker could gain access with high privilege to an SDN application and perform illegal operations.

ARP Spoofing Attack

A Man-in-the-middle attack which is also called ARP cache-poisoning. A hacker can use an ARP spoofing to infiltrate the network, sniff traffic, modify it and even stop it. This type of attack corrupts the network topology information and the topology-aware SDN applications. Poisoning can also happen through other protocols such as LLDP or IGMP.

API exploitation

The APIs of a software component might contain vulnerabilities that can allow a hacker to perform an unauthorized disclosure of information. API exploitation can also happen at the northbound interface and can lead to the destruction of network flows.

Traffic sniffing

A sniffing attack is a popular method used by hackers to capture and analyze network communication information. With sniffing, a hacker is also able to eavesdrop data from network elements or links and steal important information. Sniffing can happen anywhere where there is constant traffic. In SDN a hacker can take advantage of unencrypted communications to intercept traffic from and to a central controller. The data captured could include critical information on flows or traffic allowed on the network.

Password guessing or brute force

This attack can happen on a non-SDN element. With password guessing or brute force, an unauthorized user could gain access to the SDN.

CHAPTER 2

2.1 PROPOSED SYSTEM

In this section we will describe the methodology, RNN, SDN-based IDS architecture and the Datasets. Figure1 is the best flow representation of our proposed methodology.

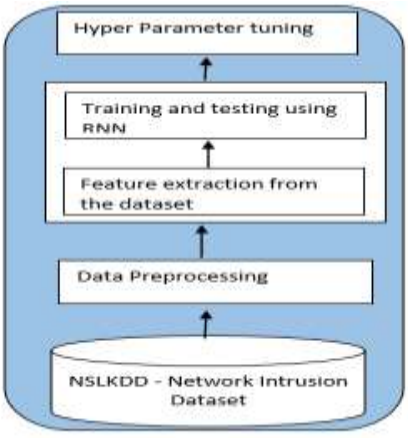

Fig 1: flow diagram for proposed work

Implement the Intrusion Detection System in Software Defined Network-based environment using Recurrent Neural Network approaches and comparing of the result with already implemented approaches i.e. ML classifiers, CNN and DNN. And check the training and testing accuracy of the model.

CHAPTER 3

PROJECT DESCRIPTION

3.1 PROBLEM DEFINITION

Implementation of Intrusion Detection System in Software Defined Network using various machine learning classifiers such as Random Forest, Decision Tree, Naive Bayes, K-Nearest Neighbour, SVM and other algorithms have been done and many papers also get published on these ML approaches to implement IDS in SDN. Recently Convolutional Neural Networks (CNN) and deep neural networks approaches have also been implemented in SDN based IDS. To overcome the network security issues in software defined network based intrusion detection systems the advanced machine learning approaches have emerged i.e. recurrent neural network, gated recurrent unit-RNN.

The main objective of doing this project was to Implement the Intrusion Detection System in Software Defined Network-based environment using Recurrent Neural Network approaches and comparing of the result with already implemented approaches i.e. ML classifiers, CNN and DNN. And check the training and testing accuracy of the model.

3.2 MODULE DESCRIPTION

3.2.1 MODULE 1-DATA PREPARATION

Searched for a predefined dataset with better features and data. Firstly, we searched for the datasets already available for the software defined network based environments. And we found a few data sets available that are listed in the next section. Used predefined dataset i.e. NSL-KDD and KDDcup 1999. After searching the datasets, we get the two datasets with efficient features and records i.e. KDDcup 99 and NSL-KDD dataset for Intrusion detection systems.

3.2.2 MODULE 2-DATA PRE-PROCESSING

Dataset had been pre-processed and then labelled. Datasets were pre-processed and labelled according to the features and class.Created the RNN model and loaded the dataset.Compared the accuracy with all ML classifiers, CNN and DNN.

NSL KDD After creating the RNN model and loading the datasets we compared the accuracy of our model with other machine learning classifiers CNN and DNN. Trained the model. Tested the model. And after that we trained and tested our model with the 10% of the KDDcup 99 and NSL-KDD datasets.

3.2.3 MODULE 3-EXPERIMENTAL EVALUATION

After creating the RNN model and loading the datasets we compared the accuracy of our model with other machine learning classifiers CNN and DNN.Trained the model. Tested the model. And after that we trained and tested our model with the 10% of the KDDcup 99 and NSL-KDD datasets.Check the result and predict the attack. After training and testing we get the desired results from our model and it predicted the attacks according to the predefined classes i.e. DoS, Probe, U2R and R2L attacks.

3.3 Recurrent Neural Network (RNN)

A recurrent neural network is a class of Artificial Neural Network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behaviour for a time sequence. Unlike feed-forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. In the proposed work we have used the CNN-GRU, CNN-LSTM and CNN-RNN methods of supervised learning for training our model. The RNN architecture has shown in figure.2.

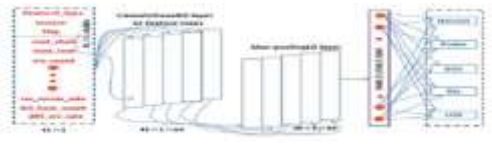

Fig 2: rnn architecture and lstm memory block

3.4 SDN-based IDS

The Software Defined Network-based Intrusion Detection System architecture is shown in figure.3. This architecture is described with three main modules i.e. Flow controller, Anomaly detector and Anomaly mitigator.

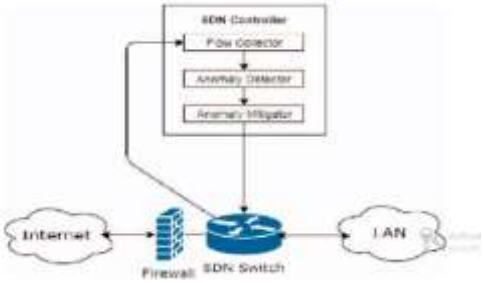

Fig 3: sdn based ids architecture

3.5 KDD cup 1999 and NSL-KDD dataset

KDD cup 1999 and NSL-KDD datasets are the main datasets used for IDS based evaluation in comparison to other listed datasets. So in proposed work we have used the NSL-KDD dataset and KDDcup 1999 dataset for training and testing our model. NSL-KDD dataset contains the essential records of the complete KDD dataset-

- Redundant records are removed to enable the classifiers to produce an unbiased result.
- Sufficient number of records is available in the train and test data sets, which is reasonably rational and enables to execute experiments on the complete set.
- The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set. In each record there are 41 attributes unfolding different features of the flow and a label assigned to each either as an attack type or as normal. The 42nd attribute contains data about the various 5 classes of network connection vectors and they are categorized as one normal class and four attack classes. The 4 attack classes are further grouped as DoS, Probe, R2L and U2R. The description of the attack classes is shown in Table I.

CHAPTER 4

CONCLUSION AND FUTURE ENHANCEMENTS

4.1 CONCLUSION

In the proposed paper, we present an Intrusion Detection System in the Software Defined Network environment using the Recurrent Neural Network. Our proposed model had shown that the Recurrent Neural Network outperforms other state-of-the-art algorithms with an accuracy of 99% and while comparing with other machine learning classifiers such as SVM, NB, LR, RF, AB and KNN, no machine learning approach approach and DNN gave the accuracy more than 92%. Our scheme uses the features of the NSL-KDD dataset and KDDcup 1999 dataset for training and testing our model and to detect the attacks such as DOS, Probe, U2R and R2L in the SDN-based environment. This makes the model more computationally efficient for real time detection. In addition, the network performance evaluation showed that our proposed approach has better accuracy and results in comparison to other approaches. Therefore, it is practical for implementation under the context of SDN.

4.2 FUTURE ENHANCEMENTS

In the future, we will optimize our model and use other features to increase the accuracy to detect more attacks in the software defined network based environment. We will also try to implement our approach in a distributed manner to reduce the overhead on the controller and we will also implement our proposed approach in the 5G networks with more efficiency and attack detection in the network using more approaches for better results.

CHAPTER 5

REFERENCES

[1] Definition of SDN available on – www.opennetworking.org/sdn.definition/.

[2] ieeexplore.ieee.org/stamp/stamp.jsp? arnumber = 8025644.

[3] Nisharani Meti, Narayan D.G and V.P. Baligar, 2017 - "Detection of DDoS attacks using Machine Learning algorithms in SDN", IEEE-978-1-5090-6367-3.

[4] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi and Mounir Ghogho, 2016 - "Deep Learning approach for Network Intrusion Detection in Software Defined Network ", IEEE-978-1-5090-3837-4

[5] Jiaqi Li, Zhifeng Zhao, Rongpeng Li, 2015 – "A Machine Learning based Intrusion Detection System for Software Defined Network in 5G", IET Research Journals.

[6] Ahmed Al Eroud, Izzat Alsmadi, 2016 – "Identifying Cyber-attacks on Software Define Network", available on - http://dx.doi.org/10.1016/j.jnca.

[7] Lorenzo Fernandez Maimo, Angel Luis Perales Gomez, Felix J. Garcia Clemente, Manuel Gil Perez and Gregorio Martinez Perez, 2018 – "A Self-Adaptive Deep Learning based System for anomaly detection in 5G networks", IEEE Members - IEEE.

[8] Ping Wang, Hsiao-Chung Lin, Wen-Hui Lin, Kuo-Ming Chao, Chi-Chun Lo, 2016 – "An efficient flow control approach for SDN-based network threat detection & migration using support vector machine", IEEE-978-1-5090-6119-8.

[9] CNN in SDN available on - https://www.semanticscholar.org/paper/Applying-convolutional-neural-network-for-network-Vinayakumar-Soman/.