Real Time IDS using SVM Machine Learning Model

Hrithik Gaikwad
Department of Information Technology
Vidyalankar Institute of Technology
Mumbai,India
hrithik.gaikwad@vit.edu.in

Suraj Balvanshi
Department of Information Technology
Vidyalankar Institute of Technology
Mumbai,India
suraj.balvanshi@vit.edu.in

Ashutosh Bist
Department of Information Technology
Vidyalankar Institute of Technology
Mumbai,India
ashutosh.bist@vit.edu.in

Prof. Yash Shah
Department of Information Technology
Vidyalankar Institute of Technology
Mumbai,India
yash.shah1@vit.edu.in

***Abstract*—Improving Network security is one the major concern of any organization. Researches are conducted to tackle this eminent problem which can impact the availability, confidentiality, and integrity of critical information of any organization. IDS is one such tool used to keep critical information secure. In this paper, we present a RT-IDS system which uses Machine Learning techniques. Using Machine Learning Techniques, results in better efficiency, reduced false positive rate and higher detection of novel attacks, as compared to Traditional IDS.**

Keywords—RT-IDS, IDS, Machine learning, Real Time IDS, ML, Security, Network Security, Networking, SVM, State Vector Machine, Real Time Intrusion Detection System, Intrusion Detection System

I. INTRODUCTION

Internet services have gotten crucial for businesses just as for people. With the expanding dependence on network services, the availability, confidentiality, and integrity of critical information have become increasingly compromised by remote intrusions. Organizations are compelled to strengthen their systems against malevolent activities and network threats. In this manner, an organization should utilize at least one or more security devices like a firewall, antivirus software, or an IDS to shield critical information from hackers.

Depending on a firewall alone isn't adequate to keep a corporate organization from a wide range of network intrusions. This is on the grounds that a firewall can't safeguard the organization against intrusions on open ports, as it is needed for network service. Henceforth, an IDS is typically introduced to supplement the firewall. An IDS gathers data from an organization or PC framework and examines the data for the occurrence of network intrusions. An organization's IDS scans network information and gives a warning to the network administrator when it distinguishes malicious activity on an open port.

Intrusion detection systems are of two types: host-based and network-based intrusion detection. Here we are focused on Host-based detection, which captures and analyses network data at the host system itself. This paper focuses on RT-IDS, where the incoming data is captured online and the detection result is reported to the system administrator, who can stop the live attack. Packet captured extract network features this feature is further store and pre-processed and passed to the ML model which uses this information to classify the network packets this happens in a continuous loop of 2 seconds.

II. RELATED WORK

Intrusion detection is on a very first step of classification. We might want to distinguish the smallest conceivable pattern of data which can help identify the type of connection. We present the idea of utilizing data gain to distinguish the connection. We likewise depict momentarily the ideas of machine learning procedures which can be utilized to detect attack [4].

In the pre-processing stage, we utilize a packet sniffer, which is worked with Jpcap library, to extricate network packet data. packet data is divided by associations between each pair of IP. This classification stage comprises of two parts which are preparing and testing network information with Machine Learning. In the classification stage, we train the chose AI calculation as an IDS model by utilizing a bunch of network records with realized answer classes. In light of the prepared IDS model, the grouping strategy can arrange the information in each record into ordinary network movement or fundamental assault types [1].

This paper has shown that high accuracy might be kept up while decreasing bogus positives utilizing the proposed model made out of SVMs, decision trees, and Naïve Bayes. To start with, the SVM is prepared binary characterization added to the dataset to indicate if the case is an attack or normal traffic. Second, attack traffic is steered through a choice tree for grouping. Third, Naïve Bayes and the choice tree will at that point vote on any unclassified attacks. Future work is to compose this model as a Java class with the end goal that it very well might be applied in different systems or applications. Further future work is to test this model on other organization traffic informational collections for additional top to bottom investigation [2].

We came across a paper which uses snort which act has an open sources network monitoring tool scan network with and stores it into a database and use machine learning to help in detection of DoS attack they use classification and different algorithm and use best model fit for the outcome [5]

III. PROPOSED METHODLOGY

We have proposed a program that will run on host machine which will listen to all the network traffic going through it and predict whether the connection is safe or not. This system is implemented in following 6 Phases.

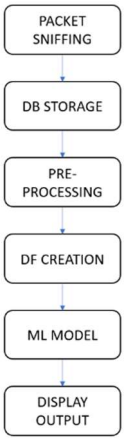

Figure 1. Six Phases

1st Phase (Packet sniffing) starts with network packet sniffing program which is return in python. The network packets which are sniffed are formatted respective with source IP addresses, destination IP address, protocol name and network features.

2nd Phase (DB Storage) in this phase the network packet sniffed is in raw form. We extract the required information and push this data in a Database for further pre-processing.

3rd Phase (Pre-processing) in this phase, the network packets are aggregated based on Source IP address and Destination IP address and these packet data will be fetched from the database.

4th Phase (DataFrame Creation). This aggregated data in DataFrame will represent the network traffic flowing through the host's end. This DataFrame becomes the input for the Machine Learning model which is the 3rd phase (ML model).

5th Phase has two sub phases A) Training: Here Machine Learning model which is trained on previously captured data by sniffer program. B) Implementing: The trained SVM model is used to predict whether its and attack or not and a new column is attached.

6th Phase (GUI): The model results are displayed on GUI which is built on python Tkinter. Were the attack packets are classified as red and normal packets are shown colorless.

These 6 phases are run in interval of every 3 sec on average which is time as 2 sec for sniffing the packets and less than 1 second for ML model to predict. These steps are iterative and executed in real time, predicting and displaying the result so.

A. System Desgin

The approach for the program is to divide it into two different parts. It started with creation of a Packet sniffer which will be used for the creation of database and a GUI for displaying result in the IDS System.

a. Packet Sniffing Design

This is the most important part of the project. Here we use python programming language and build a sniffer which will scan to the network using socket in every 2 second. Here the received network packets are in a raw form. The raw packet data is process to get the header information of packet. This header information is in binary bits format, which are then converted to decimal number format using string manipulation and this decimal number conversion and the result is passed to an if else condition to extract the required data and store it in SQL data base for pre-processing.

b. GUI

Python has its own library viz. Tkinter. Tkinter is provide a basic GUI functionality which fulfil the system need. The GUI, of the program contains separate frames for Control Buttons and the Connection Table. Control Buttons will change the value of global flag which determines whether to run sniffer or not. Connection Table displays the data of ongoing traffic though the machine.

When the ML program classify the network packets as attack, it turns the corresponding connection red colour to notify end user.

B. Dataset creation

We use a packet sniffer to scan the network packet to get information including IP header, TCP header, UDP header, and ICMP header from each packet. This packet information is partitioned and aggregating between each and every pair of IP address (source IP and destination IP) and forms the record in every 2 sec. Each record key value of data features considered as signature features representing the main characteristics of network data and activities. We performed use find key features that define the signatures of normal and. attack network traffic. We select 12 essential features for our RT-IDS approach. The features along with the data type and the information gain value. We run sniffer program for 6 continues days to collect all the normal packets and attack data. The data collected was formatted and stored in a combined csv. This csv data is used to train ML model. The following table entries are extracted from network.

No.	Feature description	Data typ
1	Number of TCP packets	Integer
2	Number of TCP source port	Integer
3	Number of TCP destination port	Integer
4	Number of TCP fin flag	Integer
5	Number of TCP syn flag	Integer
6	Number of TCP push flag	Integer
7	Number of TCP ack flag	Integer
8	Number of TCP urgent flag	Integer
9	Number of UDP packets	Integer
10	Number of UDP source port	Integer
11	Number of UDP destination port	Integer
12	Number of ICMP packets	Integer

Figure 2. Network packets attribute

C. *Machine Learning*

We have used SVM model to design are Machine learning model. A support vector machine (SVM) is a supervised machine learning model which is used for classifying result in two-group in classification problems. Here SVM is uses to predicts whether the packet scanned is normal or attack. There are 2 part in Machine Learning implementation which are Training and Testing Phase.

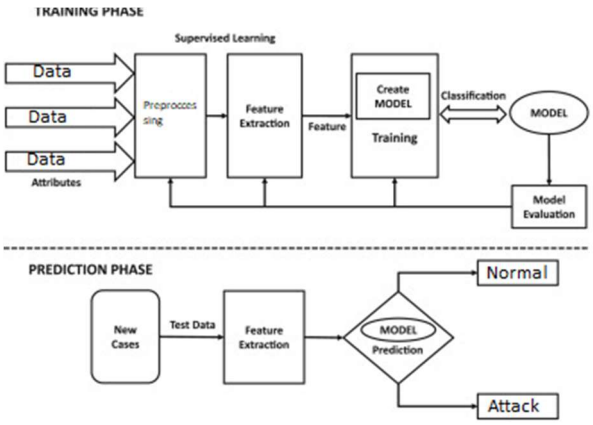

Figure 3. ML Phases

a. Training Phase: In this Phase we use 3 attack tools Etthercap, Synflood metaspoilt console and hulk phase to create dataset. The total table entries with noted were 34564. Where 11821 normal label entries and 23643 attack label entries, is given to the SVM ML model for training.

b. Prediction Phase: Trained ML model is use to predict result on unknown tool to see if it can accurately predict attack.

IV. RESULT

We have successfully developed a program the sniffs packet from the host machine, stores the packet data with the corresponding information into the database. This data contains IPv4 TCP, UDP and ICMP information.

The ML model was accurately able to distinguish between normal and attack pacts with 97.34% accuracy with known and unknown dos attack tool respectively the accuracy didn't drop.

Table 1. SVM Results

SVM	Precision	Recall	F1
	0.9423	0.9543	0.9435

The overall results of the ML model and network packets flow data is displayed on the GUI. In an interactive fashion as follows

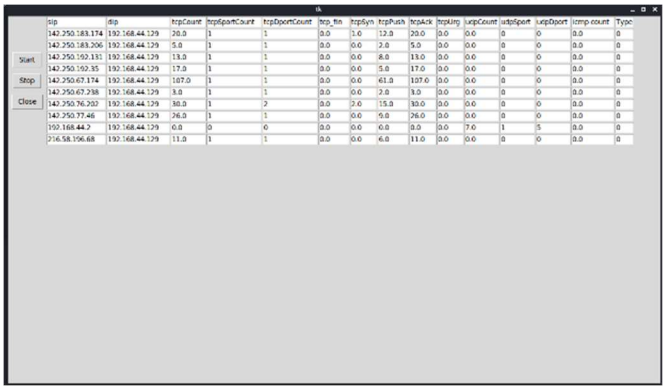

Figure 4. GUI Normal packets

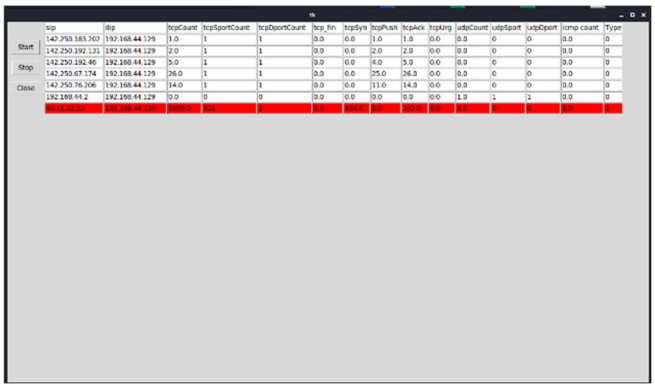

Figure 5. GUI DOS Attack

V. CONCLUSION

We implemented a SVM based RT-IDS which has overall efficiency higher than the Traditional IDS and able to overcome their shortcoming. RT-IDS system sniff network data and extract network parameter, stores that into a database which is further preprocessed by a python script. The processed data is now given to the trained Machine learning model to predict the outcome of network data. Outcome

prediction is displayed on a GUI. We use 3 different Dos attack tools to train and 3 unknow tool to train test the model and were successfully able to detect DoS with low false positive and high accuracy. Our IDS is a simple real-time host-based IDS that can efficiently detect and can also classify network data into two categories which are normal and Denial of Service (DoS).

Further we can implement the same system for different network attack such as Probe or Integrate them into a single IDS with proposed 12 relevant features or more in order to detect unknown attack types in a swift pattern.

REFERENCES

[1] PHURIVIT SANGKATSANEE, NARUEMON WATTANAPONGSAKORN, CHALERMPOL CHARNSRIPINYO, 2011. Practical real-time intrusion detection using machine learning approaches. Computer Communications, Volume 34, Issue 18, Pages 2227-2235, ISSN 0140 3664.

[2] K. GOESCHEL, 2016. Reducing false positives in intrusion detection systems using data- mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. SoutheastCon 2016, Norfolk, VA, pp. 1-6, doi: 10.1109/SECON.2016.7506774.

[3] B. ZHANG, Y. YU AND J. LI, 2018. Network Intrusion Detection Based on Stacked Sparse Autoencoder and Binary Tree Ensemble Method. 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, pp. 1-6, doi: 10.1109/ICCW.2018.8403759.

[4] LIU, H., LANG, B., 2019. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Appl. Sci. 2019, 9, 4396.

[5] Snort-Based Smart and Swift Intrusion Detection System Indian Journal of Science and Technology,Vol 11(4), DOI: 10.17485/ijst/2018/v11i4/120917, January 2018ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645

[6] KUNAL AND M. DUA, 2019. Machine Learning Approach to IDS: A Comprehensive Review. 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 117-121, doi: 10.1109/ICECA.2019.8822120

[7] HALIMAA A. AND K. SUNDARAKANTHAM, 2019. Machine Learning Based Intrusion Detection System. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 916-920, doi: 10.1109/ICOEI.2019.8862784.

[8] V. V. KUMARI AND P. R. K. VARMA, 2017. A semi-supervised intrusion detection system using active learning SVM and fuzzy c-means clustering. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, pp. 481-485, doi: 10.1109/ISMAC.2017.8058397.

[9] ZAMANI, MAHDI., 2013. Machine Learning Techniques for Intrusion Detection. Cite as: arXiv:1312.2177 [cs.CR].

[10] RAVI KIRAN VARMA P AND VALLI KUMARI V, 2012. Feature Optimization and Performance Improvement of a Multiclass Intrusion Detection System Using PCA and ANN. International Journal of Computer Applications, vol. 44, no. 13, pp. 4-9.

[11] S. X. WU AND W. BANZHAF, 2010. The use of computational intelligence in intrusion detection systems: a review. Applied Soft Computing, vol. 10, no. 1, pp. 1-35.

[12] LISHA HU, SHUXIA LU AND XIZHAO WANG, 2013. A new and informative active learning approach for support vector machine. Information Sciences, vol. 244, pp. 142-160, 2013.

[13] S. J. HORNG, M. Y. SU, Y. H. CHEN, T. W.KAO, R. J. CHEN, J. L. LAI, AND C. D. PERKASA, 2011. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications, 38(1), 306-313, 2011.

[14] Z. LIU, 2011. A method of SVM with normalization in intrusion detection. Procedia Environmental Sciences, 11, 256-262.

[15] M. C. BELAVAGI, ANDB. MUNIYAL, 2016. Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89, 117-123