Network Intrusion Detection System

Sarthak Pal
2K18/SE/116
sarthakpal07@gmail.com

1- Abstract

In today's digital world, we all use the Internet and connect to a network, but all the data we send or receive, is safe? Some kind of attack is present in network packets that might give access to the computer's private information to the hacker. We cannot see and tell whether a network is safe to connect with or not, so we made a Network Intrusion Detection Model predict whether these network packets are safe or some attack is there on the packet. We use Random Forest Classifier to obtain the maximum accuracy, and then to test our model in real-time, we have created a packet sniffer which would sniff out network packets and convert them into required features and then test it in our model to predict the legitimacy of the network packet.

2- Keywords

Some related keywords are Intrusion Detection System, Network packets, packet sniffer, Random Forest Classifier.

3- Introduction

Due to the web's general uses, electronic attacks on organizations and information systems of the monetary associations, military, and energy areas are expanding. Different intruders and hackers assault enormous sites of any association. The data of government and private associations might be spilled or harmed by unapproved clients. Intruders can have numerous structures, for example, viruses, spyware, worms, malicious logins, spam ware. Data security is a significant aspect of securing critical information about any association. The associations need security applications that adequately shield their organizations from malicious assaults and abuse. The intrusion detection system can identify the intrusions and shield the data framework from security infringement. The intrusion detection system distinguishes intruders and makes a move against the intruder. It is utilized to recover the data by fixing the association's harm brought about by an unapproved client and recognizing the malicious utilization of PC and PC organization. It identifies admittance to an un-approved client, the infringement of security, and finds illegal clients. There are two sorts of intrusion detection strategies, viz., misuse identification, and anomaly. Misuse recognition is information or example-based, though anomaly identification is behavior-based. Misuse location is dependable for identifying known assaults with low false positives. The misuse recognition method cannot distinguish the new assault. Anomaly identification procedure can recognize new assault with a high positive error. Existing intrusion detection systems have a high detection rate, though they experience high false alerts' ill effects. The undertaking of decreasing false positives is very vital for the intrusion detection system.

Since the system has the upside of finding helpful information from datasets, different Machine learning approaches are implemented. These methodologies can diminish false positives. Bayes principle, Artificial Neural Network, Hidden Markov Model, Bayesian Belief Network, Genetic Algorithm, and Association of rules and

Bunching strategies for machine learning are generally used to execute an intrusion detection system. The mix of various base Machine learning calculations is called an ensemble technique. In writing a study, it is discovered that an Ensemble technique for Machine learning assists with diminishing false-positive rates. There are three principal strategies to join fundamental Machine learning classifiers viz., Bagging, Boosting, and Stacking. In this paper, the Bagging group technique for machine learning is proposed to execute an intrusion detection system. NSL_KDD dataset and Defense Advanced Research Projects Agency (DARPA) datasets are broadly utilized as preparing and testing datasets for the intrusion detection system. The NSL_KDD dataset gives 42 features dataset to prepare and test the intrusion detection system. However, every one of the 42 features in the dataset is not essential and needed for training and testing reasons. If all features are utilized to prepare intrusion detection systems, model structure time is expanded. The critical feature selection is a whole cycle in intrusion detection systems. In light of this examination, significant features of the NSL_KDD dataset are physically chosen, which improves the order exactness.

4- Related Work

KDD [1]: This is the essential paper we are utilizing to figure out how to change over TCP dump data to the 42 features in the KDD data set.

Palo Alto [2]: Palo Alto has lately delivered six different datasets. Four of these datasets are more malware and benign projects for AI. They likewise had network traffic information and a named dataset for far off assaults to assume control over the command line/shell.

Salient Feature Extraction [3]: This paper is like the KDD Data Mining paper as it plates the features obtained in making the informational collection from crude TCP dumps. This paper focused on which of the 42 features of the KDD Cup set to end up being the most significant for the four classes of assaults, and they had gotten better outcomes for every classification when particular highlights were pruned.

Neural Network [4]: Neural Networks are presently one of the most intense classifiers in other branches of knowledge, for example, picture classification. This paper talks about different ways of how neural networks can be utilized for Anomaly Detection.

Bayesian Networks [5]: This paper is a lot like the Salient Feature paper that examines the best features to choose for machine learning. Additionally, it utilizes the Bayesian Network, which basically is an improved Bayes Classifier. They are Probabilistic Directed Acyclic Graph (DAG) models and can be perused utilizing DFS to figure the most likely occasion.

PHAD [6]: Bundle Header Anomaly Detection (PHAD) time-based convention anomaly locator for network packets. To apply time-based displaying to anomaly recognition with explicit training and trials, an inconsistency score = t*n/r is determined, where n is the occasions a packet field is seen during the training time frame, and r is the number of different estimations of a specific packet field saw during the training period, and where t is the time since the last anomaly.

5- Dataset

The experimental arrangement is isolated into two stages. In the principal stage, the NSL_KDD dataset is pre-processed. In the pre-processing stage, useful features are chosen. Feature selection is an essential information pre-preparing step to reduce the component of the dataset. A decrease in the measurement of the dataset prompts a better reasonable model. The NSL-KDD dataset has proposed 41 highlights to implement an intrusion detection system. Each of the 41 highlights is not needed to implement an intrusion detection system.

In this paper, we have included 16 features for experimental work that would give high accuracy with low false-positive rates. These features are: -

src_bytes - Number of data bytes transferred from source to destination in single connection.
count - Number of connections to the same destination host as the current connection in the past two seconds.
service - Destination network service used.
srv_count - Number of connections to the same service (port number) as the current connection in the past two seconds.
protocol_type - Protocol used in the connection.
diff_srv_rate - The percentage of connections that were to different services, among the connections aggregated in count.
same_srv_rate - The percentage of connections that were to the same service, among the connections aggregated in count.
flag - Status of the connection – Normal or Error.
dst_bytes - Number of data bytes transferred from destination to source in single connection.
srv_serror_rate - The percentage of connections that have activated the flag s0, s1, s2 or s3, among the connections aggregated in srv_count.
logged_in - Login Status: 1 if successfully logged in; 0 otherwise.
duration - Length of time duration of the connection.
lnum_compromised - Number of "compromised" conditions.
wrong_fragment - Total number of wrong fragments in this connection.
is_guest_login - 1 if the login is a "guest" login; 0 otherwise.
num_failed_logins - Count of failed login attempts.

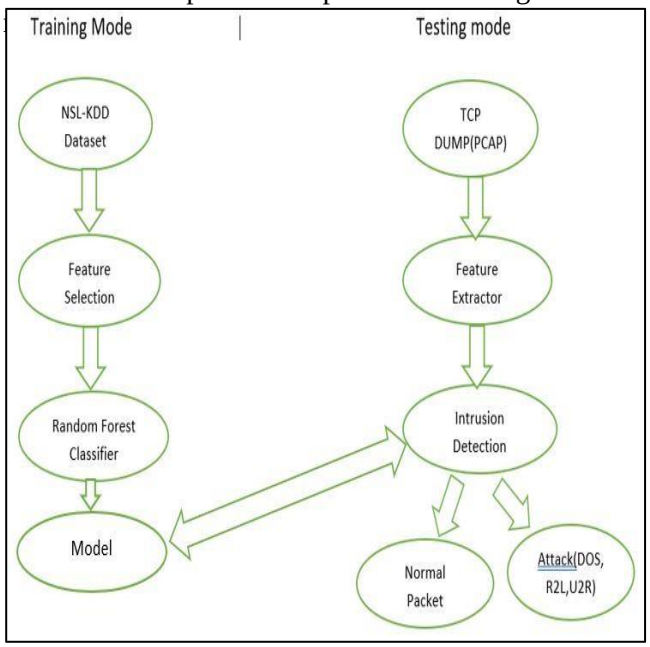
Feature Selection – We used WEKA to obtain the most correlated features. We selected a total of 16 most correlated features.

Tcpdump - Tcpdump application of Linux is used to dump all the network data transmission on all ports. Wifi is set as the default interface for the data dump, and it output the data in a cap file.

Feature Extractor- In the pcap file, Pyshark is used to analyze the data. This script basically creates all connection records in the pcap file, and then using pyshark, it scrapes all the connection and creates records according to our intrusion detection model.

Intrusion Detection - The finalized ML model is then used to predict whether the packet is regular or attack type.

Model – We divided the KDD dataset into a training and testing set of 65:35, and then we used the Random Forest Classifier of ensemble learning to train the model. We saved the model by using a pickle, which we further used to predict the packets in testing



6- Methodology and Model

The Basic Pipeline of our model is as follows: -

The attacks are broadly being classified into four types: -

1. DOS – In this attack, the intruder seeks to make a machine or network resource unavailable to its intended users by

Temporarily or indefinitely disrupting services of a host connected to the Internet.

2. R2L – This type of attack is launched by an attacker to gain unauthorized access to a victim machine in the entire network.

3. U2R – This type of attack is launched illegally to obtain the root's privileges when legally accessing a local machine.

4. Probe – A probe is an attack that is deliberately crafted so that its target detects and reports it with a recognizable "fingerprint" in the report. The attacker then uses the collaborative infrastructure to learn the detector's location and defensive capabilities from this report.

If the packet does not belong to these four types of attack, then it is classified as a regular or safe packet.

7- Results

The features were extracted from WEKA, so these features were given to our model to train with the 65% of the KDD dataset; the Random Forest Classifier was used to train our dataset, we used other methods like K-NN and Decision Trees also to train our model, but the best results were given by Random Forest Classifier. Rest 35% of the dataset was tested on the trained model, and it gave the best accuracy of 99% and had a shallow rate of false positives.

For real-time analysis of the model, we created a packet sniffer which would sniff real-time packets from the network, and then our feature extractor script decodes it to obtain the required features and then stores the data obtained in the CSV, and then our trained model tests this real-time data and classifies the packets into the five categories which are [Normal, DOS, U2R, R2L, Probe].

The best method to measure the accuracy is to calculate precision, recall, and f1 score, which are given below: -

Class	value
NORMAL	
True Positive	25479
True Negative	9291
False Positive	29
False Negative	17
DOS	
True Positive	8927
True Negative	26503
False Positive	11
False Negative	5
PROBE	
True Positive	229
True Negative	34566
False Positive	9
False Negative	11
R2L	
True Positive	119
True Negative	34674
False Positive	5
False Negative	18
U2R	
True Positive	6
True Negative	34804
False Positive	1
False Negative	5

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Precision: -

NORMAL – ~0.99
DOS – ~0.99
PROBE – 0.95
R2L – 0.96
U2R – 0.86

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

Recall: -

NORMAL – ~0.99
DOS – ~0.99
PROBE – 0.96
R2L – 0.88
U2R – 0.55

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

F1 Score:-

NORMAL – ~0.99
DOS – ~0.99
PROBE – 0.96
R2L – 0.92
U2R – 0.67

8- Conclusion

Today we live in a digital world, and every day, we tend to use the Internet and establish the connection on the network with the ports and services inside the network, so there are high risks of being attacked by a hacker who uses these kinds of attack to obtain access to monitor, so our network intrusion detector model can be used here to predict the legitimacy of the packets as it has an accuracy of 99% and could be used in an organization also for security purposes.

9- References

[1] Lee, W., & Stolfo, S. (1998). Data mining approaches for intrusion detection.

[2] Amit, I., Matherly, J., Hewlett, W., Xu, Z., Meshi, Y., & Weinberger, Y. (2018). Machine Learning in Cyber-Security-Problems, Challenges, and Data Sets. arXiv preprint arXiv:1812.07858.

[3] Staudemeyer, R. C., & Omlin, C. W. (2014). Extracting salient features for network intrusion detection using machine learning methods. South African computer journal, 52(1), 82-96.

[4] Y. Sani, A. Mohamedou, K. Ali, A. Farjamfar, M. Azman and S. Shamsuddin, "An overview of neural networks use in anomaly Intrusion Detection Systems," 2009 IEEE Student Conference on Research and Development (SCOReD), Serdang, 2009, pp. 89-92.

[5] M. A. Jabbar, R. Aluvalu and S. S. Satyanarayana Reddy, "Intrusion Detection System Using Bayesian Network and Feature Subset Selection," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, 2017, pp. 1-5.

[6] Mahoney, M. V., & Chan, P. K. (2001). PHAD: Packet header anomaly detection for identifying malicious network traffic.