Collaborative Attack Generation and Detection Using Machine Learning Techniques

Naimesh Kame[1], Sambhav Rakhe[2], Gitesh Chaudhari[3], Akash Ajnadkar[4], Shraddha Khonde[5]
[1,2,3,4] *Student, Department of Computer Engineering, Modern Education Society's College of Engineering, Pune*
[5]*Assistant Professor, Department of Computer Engineering, Modern Education Society's College of Engineering, Pune*

Abstract- **Intrusions on systems are attempts to gain access to unauthorized data with malicious intent. Intrusion Detection Systems (IDS) is a system that can detect and report such attacks. Intruders always try to evade IDS taking advantage of its impotence to detect novel attacks, combined attacks or collaborative attacks. Combined attacks are attacks on a system consisting of two or more attacks done iteratively in a loop that hide the signature of a single attack. Collaborative attacks are more sophisticated, intelligent and powerful attacks that possess the ability to merge different attacks in a single packet. These attacks can depict the behaviour of various attacks but a signature of none. Detection of such attacks is only possible with a novel IDS dataset. KDD-99 is the most common IDS dataset; we use the attacks and features available in this dataset to make our collaborative IDS dataset. We also present a host-based machine learning IDS for detecting the same.**

Index terms- **KDD99, Collaborative, Novel dataset, Intrusion Detection System.**

I.INTRODUCTION

Internet usage has seen exponential growth in the recent decade. This rapid development in technology has made protecting data even tougher. Organizations have seen an increase in data breaches that occur with more intelligent approaches than ever that takes advantage of old hardware and firmware.

Intrusions mostly occur with malicious intent, often in conquests of trial and errors with firewalls, IDS and other cybersecurity systems. Two most common IDS classes are NIDS and HIDS; which mostly use signature-based pattern analysis to detect the type of attacks. More complex IDS use techniques like multistage pattern analysis, collaborative filtering, hybrid deployment and machine learning. Thus, with evolution Intrusion Detection, attackers have also evolved new lethal tools and mechanisms that allow them to remain anonymous and cause damage.

This paper discusses the need for a novel dataset over the KDD99CUP for the detection of advanced attacks. Our work involves designing a dataset, its detailed analysis and an efficient IDS.

This paper is composed of 5 sections. Section 1 provides the introduction; Section 2 discusses past work; Section 3 elaborates our work; Section 4 reports our experimentation and result; Section 5 concludes our work and insights future possibilities.

II. LITERATURE SURVEY

IDS datasets are scarce because every dataset cannot mimic real-world malicious traffic. The most commonly available dataset is KDD99 CUP [1] dataset which is a subset of DARPA98 dataset of MIT. In detailed analysis [14] [16], there are 41 features in KDD99 dataset; including a label to determine a packet as Normal or Anomaly. This dataset has around 4.9 million tuples. The training data contains a total of 22 attack types and an additional 15 attack types in the test data only. The attacks fall under in one of four categories Probing, DoS (Denial of Service), U2R (User-to-Root), R2L (Remote-to-Local).

KDD99 is one of the most widely available datasets for intrusion detection; still, it faces some major criticism [2] [3] as follows:

1 It does not simulate real-world traffic.
2 The dataset is around 20 years old. Modern attacks cannot be detected using this dataset.
3 There is too much duplicate data.

4 Data available for U2R and R2L is very less that makes it almost impossible to detect these attacks.

5 Formal attack descriptions are unavailable; leading to an ambiguous state of assumption.

6 Some attack categories are necessarily not attacks. Example Probing is not an attack until done without authorization or malicious intention. Some attacks are tools such as SATAN, SAINT and NMAP. It is almost impossible to determine the exact use of that tool to produce a similar packet as in dataset.

7 There are 15 new attacks in testing data that are entirely independent of training data.

There are various improved versions of KDD99 dataset such as NSL-KDD [11], GureKDD [13], and 10percentKDD. KDD Extractor [12] is an open-source project that converts PCAP files to CSV and feature selection to reduce KDD99 features to 29. Reduction of features has proved to increase the accuracy of detection [17].

Many research attempts to increase the accuracy using different techniques are J48 [4], Naive Bayes [5], NBTree [6], Random Forest [7], Random Tree [8], Multilayer Perceptron [9], Support Vector Machine (SVM) [10], and Chi2 [18].

III. PROPOSED WORK

A. TECHNOLOGIES USED:

1 Scapy: It is a packet manipulation tool for computer networks, basically written in Python
It can forge or decode packets, send them on the wire, capture them, and match requests and replies. We use Scapy to generate attacks in our network and accordingly train our model.

2 Wireshark: It is a network protocol analyzer that can read/write many different capture file formats. We use Wireshark for live capture of incoming data packets and to perform analysis on the same.

3 kdd99_feature_extractor: It is a utility for extraction of a subset of KDD '99 features [1] from real-time traffic or as in our case, a .pcap file. It is compatible with Windows and Linux platforms. It is licensed under the MIT license and is available as a repository on GitHub.

B. DATASET:

Our training dataset derives its features from the KDD '99 dataset [1]. The dataset has a total of 16634 tuples and after feature selection, it has a total of 20 features.

Features
protocol_type, service, flag, src_bytes, dst_bytes, land, urgent, count, srv_count, serror_rate, srv_serror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate.

Attack Type	Attack
Single Attack	Back, Land, Neptune, Pod, Smurf
Collaborative Attack	Back -Neptune, Back- Pod, Back-Smurf

These attacks have been classified into:

1. Single Attacks:

a. LAND (Local Area Network Denial): LAND is a DOS type of attack. A large number of spoofed TCP packets are flooded on to the target system. In LAND, the destination IP address, the source IP address, source port and destination port are the same as the target machine's IP address. Thus, the target machine will keep sending itself the SYN packet replies using the same port number and fill up its buffer

b. Smurf: Smurf attack is a form of a Denial of Service (DoS) attack. Smurf involves flooding a large number of ICMP packets to the target using an IP address. Smurf can exhaust system resources by continuously sending Echo reply to every Echo request.

c. Ping of Death (POD): POD is a DOS attack that involves flooding an ICMP packet greater than 65535 bytes. POD exhausts system resources drastically because a constant effort of fragmenting and integration is required. It generally affects older Systems.

d. Neptune: Neptune is also known as SYN flooding floods the target with half-open connections, i.e. TCP SYN packets that request to open a connection but never complete it.

2. Collaborative Attacks: We define, collaborative attacks as a merger for two or more attacks in a single network packet.

a. Back-Smurf: In the Back attack, the attacker modifies the source IP of the packets so the IDS will not be able to determine the real source of an attack and it would fail to stop these incoming attack packets. Combining Back and Smurf results in a DoS attack with a randomized source IP address.

b. Back-POD: Combining Back and POD results in the attacker sending large data packets without the IDS knowing the real source because of source IP modification done on the attacker's part.

c. Back-Neptune: Spoofed TCP SYN packets flood the target system. It increases the lethality of normal Neptune attack.

C. ARCHITECTURE:

Our proposed IDS (Intrusion Detection System) uses Machine Learning models for the detection of malicious network packets. This IDS uses data that includes Collaborative attacks. A data collecting agent is used to capture the incoming network traffic here Wireshark. The captured traffic needs preprocessing and, hence it is passed to the next stage. Labels are assigned that do not include the target label. This data is testing data that is further classified using the machine learning model. If any malicious packets are detected, they are relayed to the system to generate alerts accordingly

The overall working of our system is as shown in Fig. 1.

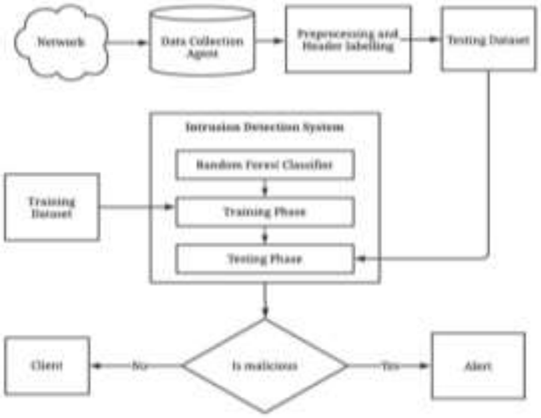

Fig: Intrusion Detection System

D. METHODOLOGY

1 Attack Generation: Scapy is used to generate attack data. Scapy is a python library and a network manipulation tool that allows creating a packet from scratch and send it to the victim.

2 Dataset Generation: This stage involves capturing data packets and obtaining the raw data. Preprocessing removes unwanted and noisy data. Extraction helps in getting features. First, the data packets are captured using a software program called Wireshark which generates a file in the .pcap format. This file is then provided to the kdd_feature_extractor [12] which generates a file in the .csv format. We then perform feature selection using the Chi-Squared test on this data and label the malicious packets to obtain information from it.

3 Attack detection: Attack detection uses a machine learning model to train on the dataset. The dataset can classify Normal and malicious traffic.

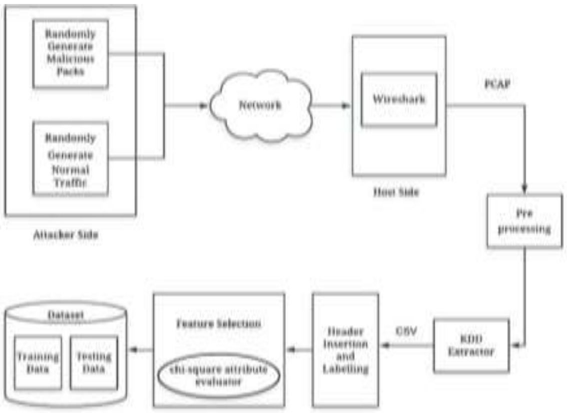
Randomly Generate Malicious Packs
Randomly Generate Normal Traffic
Attacker Side
Network
Wireshark
Host Side
PCAP
Pre processing
KDD Extractor
CSV
Header Insertion and Labelling
Feature Selection
chi-square attribute evaluator
Dataset
Training Data
Testing Data

Fig: Data Genration

Fig. 2. Depicts how we generate our datasets.

IV. RESULTS

The proposed work uses Random Forest Classifier for detection of the malicious packets. The training model achieved an accuracy of 99.847% in detecting the malicious packets.

V. CONCLUSION

This work successfully generates collaborative attacks. It creates a unique dataset by using the extractor to convert PCAP files to CSV files. It also defines a model which involves preprocessing of the input data by eliminating inconsequential tuples, feature selection using Chi2 distribution and classification of the packets using Random Forest Classifier. The proposed IDS successfully detect individual attacks, combined attacks and collaborative attacks.

VI. FUTURE WORK

The proposed system can detect other attack and attack types such as Probe, R2L, and U2R. Real-time attack detection can be possible using advanced services such as the cloud. Network deployment is possible so that host can receive only harmless traffic.

REFERENCES

[1] KDD CUP 99 available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October 2007.

[2] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 Darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.

[3] M. Mahoney and P. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection," LECTURE NOTES IN COMPUTER SCIENCE, pp. 220–238, 2003

[4] J. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[5] G. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345, 1995.

[6] R. Kohavi, "Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 7, 1996.

[7] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[8] D. Aldous, "The continuum random tree. I," The Annals of Probability, pp. 1–28, 1991

[9] D. Ruck, S. Rogers, M. Kabrisky, M. Oxley, and B. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," IEEE Transactions on Neural Networks, vol. 1, no. 4, pp. 296–298, 1990.

[10] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

[11] NSL-KDD data: http://nsl.cs.unb.ca/NSL-KDD

[12] "KDD Extractor" https://github.com/AI-IDS/kdd99_feature_extractor

[13] GureKDDCup Dataset: http://www.sc.ehu.es/acwaldap/.

[14] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," 978-1-4244-3764-1/09/$25.00 ©2009 IEEE

[15] Santosh Kumar Sahu Sauravranjan Sarangi Sanjaya Kumar Jena National Institute of Technology, Rourkela, "A Detail Analysis on Intrusion Detection Datasets," 978-1-4799-2572-8/14/$31.00_c 2014 IEEE

[16] L.Dhanabal, Dr. S.P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015

[17] Prof. Bhavin Shah, Prof. Bhushan H Trivedi, PhD, "Reducing Features of KDD CUP 1999 Dataset For Anomaly Detection Using Back Propagation Neural Network", 2327-0659/15 $31.00 © 2015 IEEE, DOI 10.1109/ACCT.2015.131

[18] H. Liu and R. Setiono, "Chi2: Feature Selection and Discretization of Numeric Attributes," Proc. Seventh International Conference on Tools with Artificial Intelligence (TAI '95), IEEE Press, Nov. 1995, pp.388-391, doi: 10.1109/TAI.1995.479783.

[19] Resul Daş, Abubakar Karabade, Gurkan Tuna, "Common Network Attack Types and Defense Mechanisms," 978-1-4673-7386-9/15/$31.00 © 2015 IEEE