Signature Analysis of UDP Streams for Intrusion Detection using Data Mining Algorithms

R.Sridevi
Asst.Prof & Head, Dept. of Information Technology
SACET
Trichy, India

DR.K.Lakshmi
Prof & Head, Dept. of Computer Science
Periyar Maniammai University
Tanjore, India

Abstract— **with the increased use of internet for a wide range of activity from simple data search to online commercial transactions, securing the network is extremely important for any organization. Intrusion detection becomes extremely important to secure the network. Conventional techniques for intrusion detection have been successfully deployed, but predictive action can help in protecting the system in the long run. Data mining techniques are being increasingly used to study the data streams and good results have been achieved over time.**

In this paper we propose to extract unique signatures from UDP data stream, apply existing mining techniques and compare results. We have used the KDD cup 1999 dataset which contains a wide variety of intrusion attacks simulated in a military environment.

Keywords- *UDP, Intrusion detection, KDD Cup dataset, Random tree, Naïve Bayes.*

I. INTRODUCTION

1.1 History of Intrusion Detection

It is very important that the security mechanisms of a system are designed so as to prevent unauthorized access to system resources and data. However, completely preventing breaches of security appear at present unrealistic. We can, however try to detect these intrusion attempts so that action may be taken to repair the damage. This field of research is called Intrusion Detection [1].

Anderson, while introducing the concept of intrusion detection, defined an intrusion attempt or a threat to be the potential possibility of a deliberate unauthorized attempt to

- access information,
- manipulate information, or
- Render a system unreliable or unusable [2].

There has been a recent awareness of the risk associated with the network attacks by criminals or terrorist as information systems are now more open to the Internet than ever before. The deployment of sophisticated firewalls or authentication systems is no longer enough for building a secure information system. In addition, most of intrusion detection nowadays relies on handcrafted signatures just like anti-virus which has to be updated continuously in order to be effective against new attacks [3].

The anomaly detection attempts to quantify usual or acceptable behaviour and flags other irregular behaviours as potentially intrusive [4]. Intrusion Detection System [IDS] plays a key role of detecting various kinds of attacks and secures the applications and networks in the pervasively connected network environment.

1.2 Basic Types of Intrusions

There are two basic techniques used to detect intruders: anomaly detection and misuse detection (signature detection) [5].

Misuse detection uses the signatures of known attacks to identify a matched activity as an attack instance, while anomaly detection uses established normal profiles to identify any unacceptable deviation as the result of an attack. Usually misuse detection is more effective against known attacks with higher true positive rate, while anomaly detection could catch new attacks but with higher false positive rates [6].

1.3 Data Mining

Data mining is the process of extracting non trivial data from huge datasets. Data mining is assisting various applications for required data analysis. Recently data mining is becoming an important component in intrusion detection system. Different data mining approaches like classification, clustering, association rule, and outlier detection are frequently used to analyse network data to gain intrusion related knowledge [7].

The process of data mining includes many steps, starting with the choice and preparation of data sources and ending with the presentation of the data mining results. In addition, it is generally accepted that the data mining is not a “one shot” process, but rather the result is obtained through iterative refinement steps of algorithm choice, parameters settings and intermediate results presentation [8].

1.4 Network Attacks

Each attack type falls into one of the following four main categories:

Denial of service (DoS) attacks:
These attacks have the goal of limiting or denying services provided to the user, computer or network. A common tactic is to severely overload the targeted system (e.g. apache, smurf, Neptune, ping of death, mailbomb, udpstorm,, SYNflood , etc.) Probing or surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network. Port scans or sweeping of a given IP address range typically fall in this category. (E.g. saint, port sweep, mscan, nmap, etc)

User-to-Root (U2R) Attacks
These attacks have the goal of gaining root or super-user access on a particular computer or system on which the attacker previously had user level access. These are attempts by a non-privileged user to gain administrative privileges.

Remote-to-Local (R2L):
This attack is an attack in which a user sends a packet to a machine over the internet, which the user does not have the access to in order to expose the machine vulnerabilities and exploit privileges which a local user would have on the computer (e.g. xclock, dictionary, guest password, sendmail, xsnoop, etc.) [9].

1.5 Naïve Bayes Model

The naïve Bayes model is a heavily simplified Bayesian Probability model. In this model, consider the probability of an end result given several related evidence variables. The probability of end result is encoded in the model along with the probability of the evidence variables occurring given that the end result occurs. The probability of an evidence variable given that the end result occurs is assumed to be independent of the probability of other evidence variables given that end results occur.
The naïve Bayes classifier operates on a strong independence assumption. This means that the probability of one attribute does not affect the probability of the other. Given a series of n attributes, the naïve Bayes classifier makes 2n! Independent assumptions. Nevertheless, the results of the naïve Bayes classifier are often correct. The work reported in examines the circumstances under which the naïve Bayes classifier performs well and why. It states that the error is a result of three factors: training data noise, bias, and variance. Training data noises can only be minimized by choosing good training data. The training data must be divided into various groups by the machine learning algorithm. Bias is the error due to groupings in the training data being very large. Variance is the error due to those groupings being too small.

The naive Bayes probabilistic model Abstractly, the probability model for a classifier is a conditional model

$$P(D/G_1,\ldots,G_n)$$

Over a dependent class variable D with a small number of outcomes or classes, conditional on several feature variables G_1 through G_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes theorem we write

$$P(D \mid G_1,\ldots,G_n) = \frac{P(D)P(G_1,\ldots,G_n \mid D)}{P(G_1,\ldots,G_n)}$$

In plain English the above equation can be written as

$$\text{Posterior} = \frac{\text{Prior} * \text{Likelihood}}{\text{Evidence}}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on D and the values of the features G_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model $(D,G_1,\ldots,G_n)$

This can be rewritten as follows, using repeated applications of the definition of conditional probability

$$P(D,G_1,\ldots,G_n)=P(D)P(G_1,\ldots,G_n \mid D$$

$$= P(D)P(G_1 \mid D)P(G_2,\ldots G_n \mid D,G_1)$$

$$=P(D)P(G_1 \mid D)P(G_2 \mid D,G_1)P(G_3,\ldots,G_n \mid D,G_1,G_2)$$

$$=P(D)P(G_1 \mid D)P(G_2 \mid D,G_1)P(G_3 \mid D,G_1,G_2)$$

$$P(G_4,\ldots G_n \mid D,G_1,G_2,G_3)$$

$$=P(D)P(G_1 \mid D)P(G_2 \mid D,G_1)P(G_3 \mid D,G_1,G_2)\ldots$$

$$P(G_n \mid D, \quad G_1,G_2,G_3,\ldots,G_{n-1})$$

Now the "naive" conditional independence assumptions come into play: assume that each feature G_i is conditionally independent

Of every other feature G_j for $j \neq i$ this means that

$$P(G_i \mid D,G_j) = P(G_i \mid D)$$

And so the joint model can be expressed as

$$P(D,G_1,\ldots,G_n) = P(D)P(G_1 \mid D)P(G_2 \mid D)P(G_3 \mid D)\ldots$$

$$= P(D) \prod^{n} P(G_i \mid D)$$

$$\prod_{i=1}^{n}$$

This means that under the above independence assumptions, the conditional distribution over the class variable *C* can be expressed like this:

$$P(D \mid G_1,\ldots,G_n)=1/Z\ [P(D) \prod_{i=1}^{n} P(G_i \mid D)]$$

Where *Z* (the evidence) is a scaling factor dependent only on $G_{1, \ldots,} G_n$, i.e., a constant if the values of the feature variables are known.

Models of this form are much more manageable, since they factor into a so-called class prior

P (D) and independent probability distributions $P(G_i \mid D)$. If there are m classes and if a model for each $P(G_i \mid D=d)$ can be expressed in terms of x parameters, then the corresponding naive Bayes model has (m − 1) + *n* x m parameters. In practice, often m = 2 (binary classification) and *x* = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is 2*n* + 1, where *n* is the number of binary features used for classification and prediction.

1.6 Decision tree and Random forest

Decision tree is one of the most powerful and simple data mining methods. The decision tree is a kind of a tree that consists of branch nodes representing a choice among a number of alternatives, and each leaf nodes representing a class of data. A simple example of decision tree is depicted in Fig. 1.

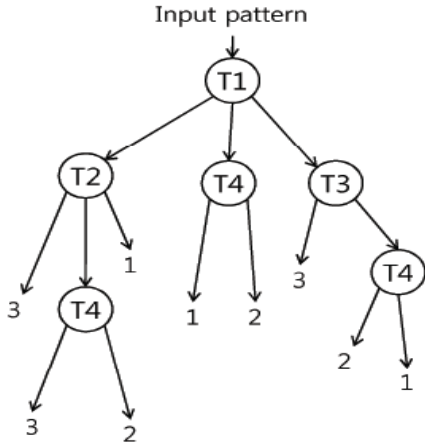

Fig.1. A Simple Decision Tree

In Fig. 1, branch nodes such as T1, T2, T3, and T4assign a class number to a input pattern by filtering the pattern down through the tests in the tree. For example, the T3 tests the input pattern down from the T1, and assigns class 3 to the input pattern or passes down to the T4. Finally, any input patterns can be categorized to the class 1, 2, or 3 when the input pattern reaches to the leaf nodes. Therefore, the decision tree is valuable to categorize the data from the large dataset [10].

Decision trees have become one of the most powerful and popular approaches in knowledge discovery and data mining, the science and technology of exploring large and complex bodies of data in order to discover useful patterns. The area is of great importance because it enables modelling and knowledge extraction from the abundance of data available. Both theoreticians and practitioners are continually seeking techniques to make the process more efficient, cost-effective and accurate. Decision trees, originally implemented in decision theory and statistics, are highly effective tools in other areas such as data mining, text mining, information extraction, machine learning, and pattern recognition [11].
Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges as to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favourable [12] but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

1.7 Neural Network
An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. Why would be necessary the implementation of artificial neural networks? Although computing these days is truly advanced, there are certain tasks that a program made for a common microprocessor is unable to perform; even so a software implementation of a neural network can be made with their advantages and disadvantages.

Advantages:

- A neural network can perform tasks that a linear program can not.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem.

Disadvantages:

- The neural network needs training to operate.

- The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
- Requires high processing time for large neural networks.

Another aspect of the artificial neural networks is that there are different architectures, which consequently requires different types of algorithms, but despite to be an apparently complex system, a neural network is relatively simple.

Artificial neural networks (ANN) are among the newest signal-processing technologies in the engineer's toolbox. The field is highly interdisciplinary, but our approach will restrict the view to the engineering perspective. In engineering, neural networks serve two important functions: as pattern classifiers and as nonlinear adaptive filters [13].

1.8 Neural networks versus conventional systems

Neural networks take a different approach to problem solving than that of conventional systems. Conventional systems use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional systems use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

Neural networks do not perform miracles. But if used sensibly they can produce some amazing results [14].

II. GOAL OF OUR WORK

We propose to extract the UDP data streams from the KDD cup data set and create a multi class dataset specifically highlighting the different intrusion threats common to UDP data streams. The signatures extracted from the dataset are used to check the classification accuracy of Naïve Bayes Algorithm, Random Tree and Neural Network. The output gives us promising results.

2.1 Dataset Used In This Research Work

The dataset used in this paper is KDD99 dataset which is suitable for the Network Intrusion Detection. The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies[15].

KDD'99 features can be classified into three groups:

1) Basic features: this category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features leading to an implicit delay in detection.

2) Traffic features: this category includes features that are computed with respect to a window interval and is divided into two groups:

a) "same host" features: examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.
b) "Same service" features: examine only the connections in the past 2 seconds that have the same service as the current connection.

The two aforementioned types of "traffic" features are called time-based. However, there are several slow probing attacks that scan the hosts (or ports) using a much larger time interval than 2 seconds, for example, one in every minute. As a result, these attacks do not produce intrusion patterns with a time window of 2 seconds. To solve this problem, the "same host" and "same service" features are re-calculated but based on the connection window of 100 connections rather than a time window of 2 seconds. These features are called connection-based traffic features.

3) Content features: unlike most of the DoS and Probing attacks, the R2L and U2R attacks don't have any intrusion frequent sequential patterns. This is because the DoS and Probing attacks involve many connections to some host(s) in a

very short period of time; however the R2L and U2R attacks are embedded in the data portions of the packets, and normally involves only a single connection. To detect these kinds of attacks, we need some features to be able to look for suspicious behavior in the data portion, e.g., number of failed login attempts. These features are called content features [16].

III. EXPERIMENTAL INVESTIGATION

The analysis of Naïve bayes algorithm, Random tree and neural networks is carried out. The results are tabulated below.

SPECIFICATIONS	NAÏVE BAYES	RANDOM TREE	NEURAL NETWORKS
Correctly Classified Instances	90.3208 %	99.8791 %	97.5502 %
Incorrectly Classified Instances	9.6792 %	0.1209 %	2.4498 %
Kappa statistic	0.3163	0.9746	0
Mean absolute error	0.0484	0.0006	0.2543
Root mean squared	0.2174	0.0246	0.0246
Relative absolute	205.3523 %	1078.1938 %	1078.1938 %
Root relative squared	198.069 %	22.4013 %	327.6132 %
Total Number of Instances	12408	12408	12409

Naïve Bayes

=== Detailed Accuracy by Class ===

TP Rate	FP Rate	Precision	Recall	Fmeasure	Class
0.901	0.007	1	0.901	0.948	Normal
0.985	0.001	0.833	0.985	0.903	teardrop
0.993	0.001	0.929	0.993	0.96	satan
1	0.095	0.073	1	0.999	nmap

Random Tree

=== Detailed Accuracy by Class ===

TP Rate	FP Rate	Precision	Recall	Fmeasure	Class
1	0.036	0.999	1	0.999	normal
0.909	0	0.968	.909	0.937	teardrop
1	0	1	1	1	satan
0946	0	0.978	0.946	0.961	nmap

Neural Network

=== Detailed Accuracy by Class ===

TP Rate	FP Rate	Precision	Recall	Fmeasure	Class
1	1	0.976	1	0.988	normal
0	0	0	0	0	teardrop
0	0	0	0	0	satan
0	0	0	0	0	nmap

IV. CONCLUSIONS

In this paper we propose to extract unique signatures from UDP data stream, apply existing mining techniques like Naïve bayes, Random tree and Neural networks .compare results with each of the technique. The research based on signature analysis based preprocessing and application of data mining techniques have shown promising results with random tree based methods able to classify at 99.88% accuracy and Neural Network based methods able to classify at 97.55% accuracy.

Further research needs to be carried out for tcp and other network packets. Improvements can be done to reduce the overall processing time.

REFERENCES

[1] A.Sundaram , "An Introduction to intrusion detection", ACM INTERNATIONAL CONFERENCE.

[2]Eugene H Spafford, The Internet Worm Program: An Analysis., In ACM Computer Communication Review, Jan 1989, 19(1), pages 17-57.

[3]Christine Dartigue, Hyun Ik Jang, and Wenjun Zeng**,** " A New Data-Mining Based Approach for Network Intrusion Detection", IEEE 2009 ,Proc. of the seventh annual communication Networks and Services Research conference.

[4]S.Kumar, "Classification and detection of computer intrusions", Ph.D thesis, Purdue Univ., West Lafayette, IN 1995.

[5]Evgeniya Nikolova, Veselina Jecheva , "Anomaly Based Intrusion Detection Using Data Mining and String Metrics", IEEE 2009 Proc. of the International conference on communications and Mobile computing.

[6]Hui Wang , Guoping Zhang, Huiguo chen and Xueshu Jiang ,"Mining Association Rules for Intrusion Detection", IEEE 2009 International conference on Frontier of Computer Science and Technology.

[7] P.Srinivasalu, et al, "Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison", 2009, Proc of International Journal of Computer Science and Network Security, VOL.9 No.6.

[8] F Angiulli, T Catarci, P Ciaccia, G Ianni, S Kimani, S Lodi, M Patella, G Santucci & C Sartori, " An integrated data mining and data presentation tool"

[9]Mrutyunjaya Panda and manas Ranjan patra**,** " Network Intrusion Detection Using Naïve Bayes", 2007, Proc of IJCSNS International Journal of Computer Science and Network Security, VOL7, No 12.

[10] Joong-Hee Lee*, et al. , "* Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System", Feb. 17-20, 2008 ICACT 2008.

[11] Lior Rokach (Ben-Gurion University, Israel) & Oded Maimon (Tel-Aviv University, Israel), " Data Mining with Decision Trees", Series in Machine perception and artificial Intelligence vol.69.

[12] Y. Freund & R. Schapire, " Machine Learning", Proceedings of the Thirteenth International conference, 148–156.

[13] Ben Kröse & Patrick van der Smagt, **"** An introduction to Neural Networks "

[14] Christos Stergiou and Dimitrios Siganos, " NEURAL NETWORKS".

[15] H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood, " Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets".

[16] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings of the 2009 IEEE symposium on computational intelligence in security and Defense Applications(CISDA2009).