New Fuzzy Logic Based Intrusion Detection System

Nitin Namdev[1] Prof. Ravindra Kumar Gupta[2] Dr. Shailendra Singh[3]
[1]M.Tech, Computer Science and Engineering
[1, 2, 3] Department of Computer Science and Engineering
[1, 2, 3] SSSIST, Sehore, India

Abstract-- In this paper, we present an efficient intrusion detection technique. The intrusion detection plays an important role in network security. However, many current intrusion detection systems (IDSs) are signature based systems. The signature based IDS also known as misuse detection looks for a specific signature to match, signaling an intrusion. Provided with the signatures or patterns, they can detect many or all known attack patterns, but they are of little use for as yet unknown attacks. The rate of false positives is close to nil but these types of systems are poor at detecting new attacks, variation of known attacks or attacks that can be masked as normal behavior. Our proposed solution, overcomes most of the limitations of the existing methods. The field of intrusion detection has received increasing attention in recent years. One reason is the explosive growth of the internet and the large number of networked systems that exist in all types of organizations. Intrusion detection techniques using data mining have attracted more and more interests in recent years. As an important application area of data mining, they aim to meliorate the great burden of analyzing huge volumes of audit data and realizing performance optimization of detection rules. The objective of this dissertation is to try out the intrusion detection on large dataset by classification algorithms binary class support vector machine and improved its learning time and detection rate in the field of Network based IDS.

Keywords: IDS, classification, KDD Data Set

I. INTRODUCTION

Following are some basic concepts on which this paper is based.

A. Types of Intrusion Detection System

Current IDSs fall into two categories:

1 Network-based Intrusion Detection System(NIDSs)
2 Host-based Intrusion Detection System (HIDSs).

These systems can be classified based on which events they monitor, how they collect information and how they reduce from the information that an intrusion has occurred. IDSs that scrutinize data circulating on the network are called Network IDSs (NIDSs), while IDSs that reside on the host and collect logs of operating system- related events are called Host IDSs (HIDSs). IDSs may also vary according to the technique by which they detect intrusions.

1) Network Based IDS

Because they only scrutinize network traffic [1], NIDS do not benefit from running on the host. As a result, they are often run on dedicated machines that observe the network flows, sometimes in conjunction with a firewall. In this case, they are not affected by security vulnerabilities on the machines they are monitoring. Nevertheless, only a limited number of information can be inferred from that gathered on the network link. Besides, widespread adoption of end-to-end encryption further limits the amount of information that can be gathered at the network interface.

Another major shortcoming of NIDS is that they are oblivious to local root attacks. An authorized user of the system that attempts to gain additional privileges will not be deleted if attack is performed locally. An authorized user of the system may be able to set up an encrypted channel when accessing the machine remotely.

2) Host Based IDS

HIDS have an ideal vantage point [1]. Because an HIDS runs on the machine it monitors, it can theoretically observe and log any event occurring on the machine. However, the complexity of current operating system often makes it difficult, if not impossible to accurately monitor certain events. There are certain difficulties faced by security tools that rely on system calls interposition to monitor a host. In addition to shortcomings resulting from an incorrect or incomplete understanding of the operating system, race conditions in the operating system make the implementation of such tools delicate. HIDSs are also confirmed with difficulties arrived from arising from potential tampering by the attacker. A secure logging mechanism is necessary to prevent logs from being erased if the attacker compromises with the machine. Even if such a mechanism is available, an attacker obtaining super user privilege on the host can disable the HIDS. If the HIDS is a user process, an attacker can simply terminate the process. If it is embedded in the kernel, the attacker can modify the kernel by loading a kernel module or by writing directly in the kernel memory. This means that an HIDS can only be trusted

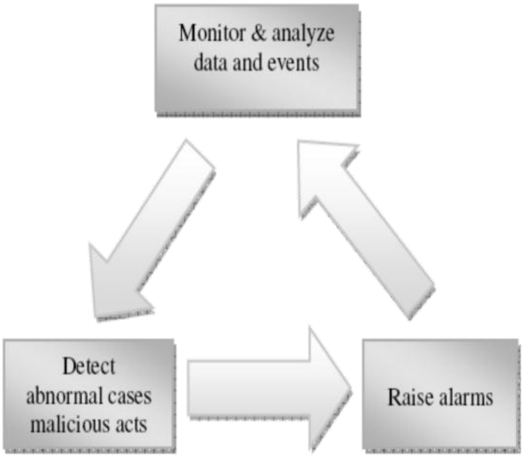

Monitor & analyze data and events
Detect abnormal cases malicious acts
Raise alarms

Fig.1: Traditional IDS framework

B. *Decision trees: -*

The well-known machine learning techniques, a decision tree is composed of three basic elements:

1. A decision node specifying a test attributes.
2. An edge or a branch corresponding to the one of the possible attribute values which means one of the test attribute outcomes.
3. A leaf which is also named an answer node contains the class to which the object belongs.

In decision trees, two major phases should be ensured:

1. Building the tree**:** Based on a given training set, a decision tree is built. It consists of selecting for each decision node the _Appropriate 'test attributes and also to define the class labeling each leaf.
2. Classification: In order to classify a new instance, we start by the root of the decision tree, then we test the attribute specified by this node. The result of this test allows moving down the tree branch relative to the attribute value of the given instance. This process will be repeated until a leaf is encountered. The instance is then being classified in the same class as the one characterizing the reached leaf.

Decision trees have also been used for intrusion detection [3]. The decision trees select the best features for each decision node during the construction of the tree based on some well-defined criteria. One such criterion is to use the information gain ratio. [2]

II. PROPOSED ALGORITHM

Classification is a form of data analysis that extracts models describing important data classes. These models also called as classifiers are used to predict categorical (discrete, unordered) class labels. This analysis can help us for better understanding of large data. Classification has numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, credit risk and medical diagnosis [4].Data Classification is a two-step process. They are: Learning Step and Classification Step

A. *Learning Step:*

In this step classification model is constructed. A classifier is built describing a predetermined set of data classes or concepts. In learning step or training phase, where classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.

This step is also known as supervised learning as the class label of each training tuple is provided. This learning of the classifier is "supervised" by telling to which class each training tuple belongs. In unsupervised learning or clustering, the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance.[4,5]

B. *Classification Step:*

In this step, the model is used to predict class labels for given data and it is used for classification. First, the predictive accuracy of the classifier is estimated. To measure the classifier accuracy, if we use the training set it would be optimistic, because the classifier tends to over fit the data i.e., during learning it may incorporate some particular anomalies of the training data that are not present in the general data set. Therefore, a test set is used, made up of the test tuples and their associated class labels., They are independent of the training tuples, from which the classifier cannot be constructed. The accuracy of a classifier on a given test set is the percentage of test tuples that are correctly classified by the classifier. The associated class label of each test tuple is compared with the learned classifies class prediction for the tuple. If the accuracy of the model or classifier is considered acceptable, the model can be used to classify future data tuples or objects for which the class label is not known [4, 6].

C. *Decision Tree Induction*

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The topmost node in a tree is the root node.

Given a tuple K, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class predicate for that tuple. Decision trees are easily converted to classification rules. The construction of decision does not require any domain knowledge or parameter setting. It can handle high dimension data. The learning and classification steps are simple and fast. It has good accuracy. Decision tree Induction algorithm can be used in many applications like medicine, manufacturing and production etc.

III. PROPOSED METHOD

A. *INPUT:*

1. A TRAINING DATA SET WITH CLASS LABELS
2. LIST OF ATTRIBUTES
3. FEATURE SELECTION CRITERIA. IT IS USED FOR SPLITTING

B. *OUTPUT:*

A FUZZY DECISION TREEPROCEDURE:

```
STEP 1:  IF ALL THE TUPLES OF DATASET
         CONTAINS YES THEN CREATE A YES
         NODE AND STOP
         ELSE IF ALL THE TUPLES OF DATASET
         CONTAINS NO THEN CREATE A NO
         NODE AND STOP
         ELSE SELECT A FEATURE AND CREATE
         A DECISION NODE
STEP 2:  SPLIT THE DATA SET D INTO SMALLER
         FUZZY SUBSETS D1, D2,…… DN
         ACCORDING TO STEP 1 CRITERIA
STEP 3:  APPLY THE ALGORITHM RECURSIVELY
         ON EACH DATA SET D1, D2… DN
```

C. *ATTRIBUTE SELECTION:*

Our proposed methodology uses greedy approach to select the best attribute. To do so the information gain is used. The

attribute with highest information gain is selected. Entropy measures the amount of information in an attribute[5].Given a collection D of c outcomes
Entropy (D) = D [-p (J) log2 p(J)]
Where; p (J) is the proportion of D belonging to class J. D is over c. Log2 is log base 2. Here; D is not an attribute but the entire sample set.
If the Entropy is 0 then all members of D belong to the same class i.e., the data is perfectly classified. If the Entropy is 1 then all members of D are totally random. The range of entropy is between0 to 1.

Gain (D, A) is information gain of example set D on attribute A is defined as
Gain (D, A) = Entropy (D) - S ((|Da| / |D|) *Entropy(Da))
Where: S is each value v of all possible values of attribute A
Da= subset of D for which attribute A has value a.
| Da| = number of elements in Da

|D| = number of elements in D.

1) KDD Data Set Description:
The DARPA/MIT Lincoln lab evaluation (IDEVAL) data set has been used to test a large number of intrusion detection systems. The data can be used to test both host based and network based systems, and both signature and anomaly detection systems. The 1998 network data has also been used to develop the 1999 KDD cup machine learning competition. The data set that we used in our experiment is the data set for KDD 99 cup machine learning competition, which is a subset of the 1998 DARPA intrusion detection evaluation data set, and is processed, extracting 41 features from the raw data of DARPA 98 data set.

For our purposes in the creation of input data for the Proposed Algorithm, we will look at the following example log entry.

Packets sent

Protocol

Source Port

Destination Port

2) Training Data Set:
The Training data set is as follows:

idsalert ,packets ,proto, sport, dport

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

Where idsalert is for IDS alert, packets is for number of packets sent, sport is for source port, dport is for destination port.

3) Testing Data Set:
The testing data set is as follows:

Ids alert packets proto sport dport

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

yes 0 tcp 1 2

yes 145 udp 148 1

no 2138 tcp 33 34

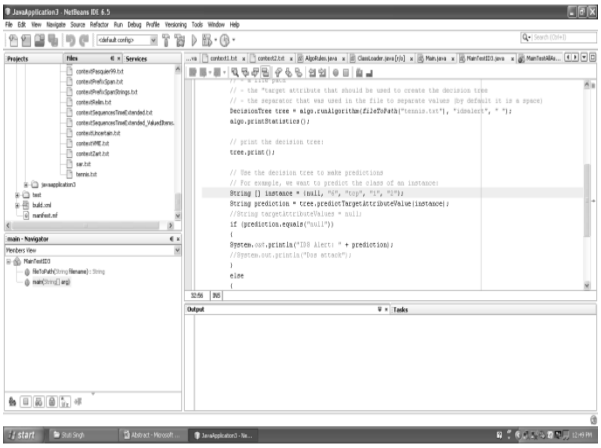
no 12 tcp 2 3

yes 230 tcp 2 120

yes 6 tcp 1 2

4) Working of proposed algorithm:
We will train our classifier with the training data set. Then we will enter the values of a tuple from the testing data set in to our program. Our classifier will tell that the particular log entry is normal or malicious. It is shown below:

When we enter the tuple no 6 from training data set in to our program as below:



```
// - the "target attribute that should be used to create the decision tree
// - the separator that was used in the file to separate values (by default it is a space)
DecisionTree tree = algo.runAlgorithm(fileToPath("tennis.txt"), "idsalert", " ");
algo.printStatistics();

// print the decision tree:
tree.print();

// Use the decision tree to make predictions
// For example, we want to predict the class of an instance:
String [] instance = {null, "6", "top", "1", "2"};
String prediction = tree.predictTargetAttributeValue(instance);
//String targetAttributeValues = null;
if (prediction.equals("null"))
{
System.out.println("IDS Alert: " + prediction);
//System.out.println("Dos attack");
}
else
{
```

5) We get the following result:

```
Debug:
Time to construct decision tree = 94 ms
Target attribute = idsalert
Other attributes = packets proto sport dport
DECISION TREE
 dport->
   3=no
   2=yes
   1=yes
   120=yes
   34=no
IDS Alert: yes
Dos attack/Probe
BUILD SUCCESSFUL (total time: 9 seconds)
```

D. Advantages of Proposed Method:

1 False Positives: A false positive occurs when the outcome is incorrectly predicted as yes when it is actually no.
False Negatives: A false negative occurs when the outcome is incorrectly predicted as no when it is actually yes.
Success Rate = (TP + TN)/(TP + TN + FP + FN)
We tested our program for the whole testing data set the success rate is 99.68%.

2 Also our method is more suitable for novelty attack detection. One can check it by using the combination of 2-3 tuples from testing data set. Even then our algorithm is providing the correct output.

IV. CONCLUSION

Achievements are as follows:

1 Reduced model redundancy by using feature selection algorithm without affecting performance.
2 Success rate is higher
3 Good for novelty attack detection

REFERENCES

[1] Litty Lionel, "Hypervisor-based Intrusion Detectio", Master of Science Graduate department of computer Science University of Torronto, 2005.

[2] http://www.mendeley.com/research/naive-bayes-vs-decisiontrees-in-intrusion-detection-systems/#page-1

[3] Srinivas Mukkamalaa, Andrew H. Sunga and Ajith Abrahamb Intrusion detection using an ensemble of intelligent paradigms‖;www.elsevier.com/locate/jnca, January 2004

[4] "Data Mining Concepts and Techniques" byJiawei Han and Micheline Kamber from Morgan Kaufman Publications

[5] Adriaan; "Introduction to Data Mining",Addison Wesley Publication

[6] Adriaan; "Introduction to Data Mining",Addison Wesley Publication
.