International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 2– No.2, June 2012 – www.ijais.org

Anomaly Detection using a Clustering Technique

Anusha Jayasimhan
M.E Computer student
Thadomal Shahani Engineering College
Mumbai,India

Jayant Gadge
Asst. Professor, Computer Department
Thadomal Shahani Engineering College
Mumbai,India

ABSTRACT
Computer networks are usually vulnerable to attacks by any unauthorized person trying to misuse the resources.. Hence they need to be protected against such attacks by Intrusion Detection Systems (IDS). The traditional prevention techniques such as user authentication, data encryption, avoidance of programming errors, and firewalls are only used as the first line of defense. But, if a password is weak and is compromised, user authentication cannot prevent unauthorized use. Similarly, firewalls are vulnerable to errors in configuration and sometimes have ambiguous/undefined security policies. They fail to protect against malicious mobile code, insider attacks and unsecured modems. Therefore, intrusion detection is required as an additional wall for protecting systems.

Previously many techniques have been used for the effective detection of intrusions. One of the major issues is however the accuracy of these systems i.e an increase in the number of false negatives. Due to the increasing amount of new and novel types of attacks, any activity which is harmful or malicious may not be identified. To overcome this issue, a clustering technique i.e Simple K Means is used to identify and detect novel attacks and also to reduce the false negative rate.

General Terms
Pattern recognition, intrusion detection

Keywords
Anomaly detection, Simple K Means, feature selection

1. INTRODUCTION
With the rapid progression of computer technology, computer violations are increasing at a fast pace. Such malevolent activities become more and more sophisticated and can easily cause millions of dollar in damage to an organization. Detecting those intrusions becomes an important issue of computer security.

Generally, there exist two main intrusion detection techniques: anomaly detection and misuse detection. Misuse detection involves the comparison of observed traffic data with a set of well defined rules that describe signatures of intrusions. If the signature of observed network traffic is not matched with any of predefined rules, it is declared as an attack. This approach can detect the recognized attacks in an efficient way with high level of accuracy. However, it suffers from its inability of identifying attacks which differ from those predefined patterns. A minor variation of an attack may normal usage patterns learned from training data. If the pattern of observed data is different from those learned normal ones, the data is classified as an attack. This approach can successfully detect novel and unseen malicious occurrences from computer users.

This paper mainly focuses on the anomaly intrusion detection technique by using a simple K means clustering approach. The rest of the paper has been organized as follows. Section 2 describes the various techniques that have been implemented in the past, Section 3 describes the various issues involved in designing the system, Section 4 describes the architecture of the system and the datasets used, Section 5 shows the results after applying the K means technique.

2. LITERATURE REVIEW
Currently, building an effective IDS is an enormous knowledge engineering task. System builders rely on their intuition and experience to select the statistical measures for anomaly detection. Experts first analyze and categorize attack scenarios and system vulnerabilities, and hand-code the corresponding rules and patterns for misuse detection. Because of the manual and ad hoc nature of the development process, current IDSs have limited extensibility and adaptability. Many IDSs only handle one particular audit data source, and their updates are expensive and slow [1].In the following section some of the techniques that have already been used in the past have been discussed.

2.1 Genetic Algorithm
Genetic algorithms were originally introduced in the field of computational biology. Since then, they have been applied in various fields with promising results. Fairly recently, researchers have tried to integrate these algorithms with IDSs.

The REGAL System [2] [3] is a concept learning system based on a distributed genetic algorithm that learns First Order Logic multi-modal concept descriptions. REGAL uses a relational database to handle the learning examples that are represented as relational tuples.

Dasgupta and Gonzalez [4] used a genetic algorithm, however they were examining host-based, not network-based IDSs. Instead of running the algorithm directly on the feature set, they used it only for the meta-learning step, on labeled vectors of statistical classifiers. Each of the statistical classifiers was a 2-bit binary encoding of the abnormality of a particular feature, ranging from normal to dangerous.

2.2 Neural Networks
The application of neural networks for IDSs has been investigated by a number of researchers. Neural networks provide a solution to the problem of modeling the users' behavior in anomaly detection because they do not require

any explicit user model. Neural networks for intrusion detection were first introduced as an alternative to statistical techniques in the intrusion detection expert system (IDES) to model . In particular, the typical sequence of commands executed by each user is learned [5]. IDSs should involve the use of pattern recognition and learning by example approaches for the following two main reasons:

• The capability of learning by example allows the system to detect new types of intrusion.

• With earning by example approaches, attack "signatures" can be extracted automatically from labeled traffic data. This basically eliminates the subjectivity and other problems introduced by the presence of the human factor.

2.3 Decision Tree

Decision tree (DT) induction is one of the classification algorithms in data mining. The classification algorithm is inductively learned to construct a model from the pre classified data set. A DT consists of nodes, leaves and edges. A node of a DT specifies an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to a possible value of edges and a possible value of the attribute in the parent node. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. To classify an unknown object, one starts at the root of the DT and follows the branch indicated by the outcome of each test until a leaf node is reached. The name of the class at the leaf node is the resulting classification. DT induction has been implemented with several algorithms. Some of them are ID3 developed by Quinlan and later on it was developed into C4.5 [6].

2.4 Fuzzy Logic

Fuzzy logic is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as "the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem"[7].

In Dickerson and Dickerson 2000 [8] the authors classify the data based on various statistical metrics. They then create and apply fuzzy logic rules to these portions of data to classify them as normal or malicious. They found that the approach is particularly effective against scans and probes.

An enhancement of the fuzzy data mining approach has also been applied by Florez *et al*[9] The authors use fuzzy data mining techniques to extract patterns that represent normal behavior for intrusion detection. They describe a variety of modifications that they have made to the data mining algorithms in order to improve accuracy and efficiency. They use sets of fuzzy association rules that are mined from network audit data as models of "normal behavior." To detect anomalous behavior, they generate fuzzy association rules from new audit data and compute the similarity with sets mined from "normal" data. If the similarity values are below a threshold value, an alarm is issued.

3. DESIGN ISSUES

A basic premise for intrusion detection is that when audit mechanisms are enabled to record system events, distinct evidence of legitimate activities and intrusions will be manifested in the audit data. Because of the sheer volume of audit data, both in the amount of network records and in the number of system features (i.e., the fields describing the network records), efficient and intelligent data analysis tools are required to discover the behavior of system activities. This leads to some drawbacks in the intrusion detection systems.

• Current IDS are usually tuned to detect known service level network attacks. This leaves them vulnerable to original and novel malicious attacks.

• **Data overload**: Another aspect which does not relate directly to misuse detection but is extremely important is how much data an analyst can efficiently analyze. That amount of data he needs to look at seems to be growing rapidly. Depending on the intrusion detection tools employed by a company and its size there is the possibility for logs to reach millions of records per day.

• **False positives (FP)**: A common complaint is the amount of false positives an IDS will generate. A false positive occurs when normal attack is mistakenly classified as malicious and treated accordingly.

• **False negatives**: This is the case where an IDS does not generate an alert when an intrusion is actually taking place (Classification of malicious traffic as normal). Hence in order for making the system as efficient and accurate as possible the following issues have been considered.

3.1 Feature Selection

Feature selection, also known as subset selection or variable selection, is a process commonly used in machine learning, wherein a subset of the features available from the data is selected for application of a learning algorithm. Feature selection is necessary either because it is computationally infeasible to use all available features, or because of problems of estimation when limited data samples (but a large number of features) are present [10]. Generally, a data set that includes a large amount of network traffic is necessary to be collected in advance for designing an intrusion detection system. The size of data collected from the network is always large. It includes a great amount of traffic records with a number of various features such as the length of connection, the type of protocol, the network service and other information. Based on this set of data, misuse detection techniques specify well defined attack signatures and anomaly detection techniques construct acceptable user behaviors.

3.2 Clustering

At the most basic level, accuracy measures how well an IDS detects attacks. There are several elements that affect the accuracy measurement. One important component is detection rate, which is the percentage of attacks that a system detects. Another component is the false negative rate, which is the percentage of anomalous data that the system falsely determines to be normal. Clustering is a data mining approach that seeks to find homogenous groups of objects based on the values of their attributes. Clustering can be viewed as a method of outlier detection where outliers are objects not located in the clusters of the data sets. Hence in the context of intrusion detection outliers may describe those activities which are intrusions or attacks. The false negative rate can thus be decreased by decreasing the number of outliers during clustering [11].

4. SYSTEM ARCHITECTURE

As explained in the previous section, one of the main issues in the IDS is to reduce the number of false negatives. In order to achieve this, patterns are constructed that represent the normal

behavior of the network traffic. The architecture of the system, to detect those patterns is shown in Fig 1:

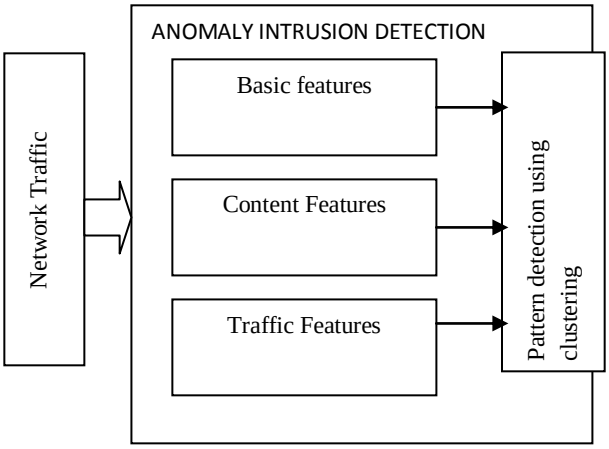

Fig 1: System architecture

The features of a network traffic can be classified broadly into 3 categories as described below [12].

4.1 Basic Features

Basic features can be derived from packet headers without inspecting the payload. This category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features lead to an implicit delay in detection.

4.2 Content Features

Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts. Content features, extracted from packet content within a connection, allow information at access level. They provide different indicators on connections status such as the number of root and access control files access, the identity of logged entity and others. Some 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.

4.3 Traffic Features

These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval. 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 **b)"same service" features**: They examine only the connections in the past 2 seconds that have the same service as the current connection, behavior, service, etc. 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.

4.4 Clustering using Simple K Means

K-means clustering is a clustering analysis algorithm that groups objects based on their feature values into K disjoint clusters. Objects that are classified into the same cluster have similar feature values. An essential problem of the K-means clustering method is to determine an appropriate number of clusters K. K is a positive integer number specifying the number of clusters, and has to be given in advance [13]. The clustering algorithm consists of the following steps:

1) Define the number of clusters K. As initial value, K =2, assuming that normal and anomalous traffic in the training data form two different clusters.

2) Initialize the K cluster centroids. This can be done by arbitrarily dividing all objects into K clusters, computing their centroids, and verifying that all centroids are different from each other. Alternatively, the centroids can be initialized to K arbitrarily chosen, different objects.

3) Iterate over all objects and compute the distances to the centroids of all clusters. Assign each object to the cluster with the nearest centroid. A distance function is required in order to compute the distance (i.e. similarity) between two objects. The distance function used is the Euclidean distance which is defined in equation 1 as follows:

$$d(x,y) = \sqrt{\left(\sum_{i=1}^{m} (x_i - y_i)^2\right)} \qquad (1)$$

where x = (x1,x2… xm) and y = (y1,y2…. ym) are two input vectors with m quantitative features.

4) Recalculate the centroids of both modified clusters.

5) Repeat step 3 until the centroids do not change any more

The k means algorithm is first trained on the dataset having both normal and anomalous traffic. The distances to the cluster centroids of the corresponding traffic class are calculated using the weighted Euclidean distance function. An object is classified as normal if it is closer to the normal cluster centroid than to the anomalous one, and vice versa. This is illustrated in Figure 2 with a two-dimensional feature space: Object P is closer to the normal cluster, therefore P is normal. This distance-based classification allows detecting known kinds of anomalies, i.e. anomalous traffic with similar characteristics as in the training datasets.

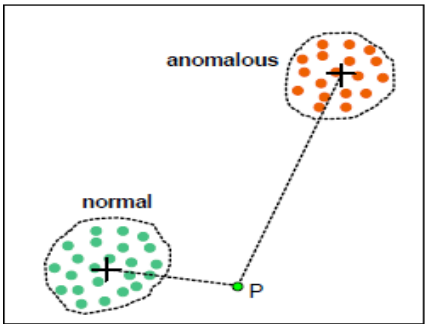

Fig 2: The clustering of P when K=2

4.5 Experimental Methodology

For the purpose of experiments, a large database called as DARPA KDD is used that contains a large volume of network

traffic connections describing TCP connections. Each connection includes 41 features plus a label of either normal or a type of attack. The content of those features are continuous, discrete, or symbolic with vary scales and ranges All attacks in DARPA Sets can be categorized into 4 classes of attacks[14]. The classes are summarized as follows.

Denial of Service (dos): Attacker tries to prevent legitimate users from using a service.

Remote to Local (r2l): Attacker does not have an account on the victim machine, hence tries to gain access.

User to Root (u2r): Attacker has local access to the victim machine and tries to gain super user privileges.

Probe: Attacker tries to gain information about the target host.

5. RESULTS

For training the system a part of the DARPA KDD dataset is considered which consists of 12190 records of the network connection out of which 6503 records are of normal non-malicious category, 0 connections of land, 4041 connections of neptune,81 connections of warezclient,321 connections of ipsweep,87 connections of teardrop,273 connections of portsweep,30 connections of pod,12 connections of guess_passwd,145 connections of nmap,333 connections of satan,258 connections of smurf,5 connections of multihop,83 connections of back,2 connections of ftp_write,4 connections of buffer_overflow,2 connections of imap,2 connections of phf,3 connections of rootkit,5 connections of warezmaster.

For the purpose of testing a part of KDD test dataset is considered that consists of 5073 records. Out of the entire record set 2569 records are of the class normal,5 records of the class land,1716 of the class neptune,33 of the class warezclient,151 records of the class ipsweep,32 records of the type teardrop,136 connections of the portsweep,17 of the type pod,19 of the class guess_passwd,56 records of the class nmap,166 records of the class satan,94 of the class smurf,10 records of multihop,33 connections of back,4 connections of ftp_write,4 records of the class buffer_overflow,1 connection of imap,2 connections of phf,10 connections of rootkit and 15 records of warezmaster. Once the system has been trained, it can be tested for it's performance on different sets. The different sets include whole training set itself, splitting the training dataset and providing a completely different test dataset. Based on the above records the results are obtained separately for the system as shown in the Table 1.

Table 1: Testing the system on different datasets

Dataset	# of Instances classified as anomaly	# of Instances classified as normal	Root mean squared error
Training dataset	4619	7571	18.952
User supplied test set	2020	3053	18.932
66% split on training set	1586	2559	14.335
50% split on training set	3776	2319	11.567

Fig 3 shows the graph generated after training the dataset. There are two clusters, cluster 0 that represents an anomaly and cluster 1 representing a normal record. The instance number has been plotted on the X axis whereas the label of the record (normal/attack) has been plotted on the Y axis.

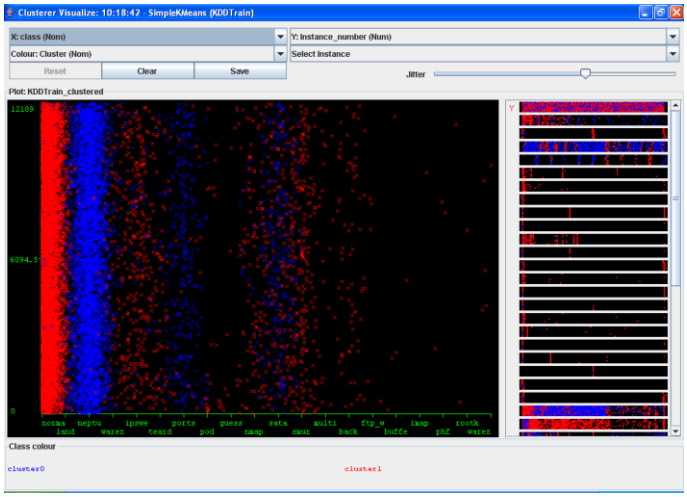

Fig 3: Visualization of the clusters

6. CONCLUSION AND FUTURE WORK

The accuracy of any intrusion detection system is determined by the detection rate, false positive rate and the false negative rate. The signature based intrusion detection systems store the patterns of attacks and hence cannot detect novel attacks. To overcome this drawback, anomaly detection system is used, which stores patterns of normal activity. While it reduces the false negative rate, anomaly detection systems have a disadvantage of a high false positive rate. It means that even if a network activity is normal a false alarm might be raised. Hence, by creating a hybrid system where both signature and anomaly based intrusion detection systems can be combined together, we can reduce the FP rate and the false negative rate.

The future enhancements of the above system could be to extract the patterns of normal records and use them as real time IDS. The system can be combined with detectors and sensors to monitor the incoming network traffic and detect any packet that does not match the stored patterns.

7. REFERENCES

[1] Lee,Salvatore J. Stolfo," A framework for constructing features and models for intrusion detection systems," ACM Transactions on Information and System Security, Vol. 3, No. 4, November 2000, Pages 227–261.

[2] Neri, F., "Comparing local search with respect to genetic evolution to detect intrusion in computer networks", In Proc. of the 2000 Congress on Evolutionary Computation CEC00, La Jolla, CA, pp. 238243. IEEE Press, pp 16-19 July, 2000.

[3] Neri, F., "Mining TCP/IP traffic for network intrusion detection", In R. L. de M'antaras and E. Plaza (Eds.), Proc. of Machine Learning: ECML 2000, 11th European Conference on Machine Learning, Volume 1810 of Lecture Notes in Computer Science, Barcelona, Spain, pp. 313-322,May 31- June 2, 2000.

[4] Dasgupta, D. and F. A. Gonzalez, "An intelligent decision support system for intrusion detection and response",In Proc. of International Workshop on Mathematical Methods, Models and Architectures for

Computer Networks Security (MMM-ACNS), St.Petersburg. Springer-Verlag, 21-23 May, 2001.

[5] Debar, H., Becker, M., and Siboni, D., "A neural network component for an intrusion detection system", IEEE Computer Society Symposium on Research in Security and Privacy, Los Alamitos, CA, pp. 240–250, Oakland, CA, May 1992.

[6] Sandhya Peddabachigaria, Ajith Abrahamb, Crina Grosanc,Johnson Thomasa," Modeling intrusion detection system using hybrid intelligent systems", Journal of Network and Computer Applications, June 2005

[7] G. J. Klir, "Fuzzy arithmetic with requisite constraints", Fuzzy Sets and Systems, 1997.

[8] Dickerson, J. E. and J. A. Dickerson, "Fuzzy network profiling for intrusion detection", In Proc. of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, pp. 301-306. North American Fuzzy Information Processing Society (NAFIPS),July 2000.

[9] G. Florez, SM. Bridges, Vaughn RB, "An improved algorithm for fuzzy data mining for intrusion detection", Annual Meeting of The North American Fuzzy Information Processing Society Proceedings, 2002.

[10] http://www.wikipedia.com visited on 02/04/2012

[11] Wenke Lee , Salvatore J. Stolfo , Philip K. Chan , Eleazar Eskin , Wei Fan , Matthew Miller , Shlomo Hershkop , Junxin Zhang," Real time data mining-based intrusion detection ,2001

[12] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani," A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings on the 2009 IEEE Symposium on Computation Intelligence in Security and Defense Application, July 2009, pp 1-6

[13] Gerhard Münz, Sa Li, and Georg Carle, " Traffic anomaly detection using k-means clustering" , In Proceedings of Leistungs-,Zuverlässigkeits-und Verlässlichkeitsbewertung von Kommunikationsnetzen und Verteilten Systemen, GI/ITG-Workshop MMBnet, September 2007

[14] H. Günes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood," Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets", Proceedings of the Third Annual Conference on Privacy Security and Trust PST2005 ,2005,pp 3-5