International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

An Intrusion Detection Model for Detecting Type of Attack Using Data Mining

Amruta Surana[1], Shyam Gupta[2]

[1]ME Student, Siddhant College of Engineering, Pune, Maharashtara, India

[2]Associate Professor, Department of Computer Engineering, Siddhant College of Engineering, Pune, Maharashtra, India

Abstract. ***Intrusion detection systems (IDS) are important elements in a network's defenses to help protect against increasingly sophisticated cyber attacks. This project objective presents a novel anomaly detection technique that cans b e u s e d to detect previously unknown attacks on a network by identifying attack features. This effects -based feature identification method uniquely combines k-means clustering; Naïve Bayes feature selection and 4.5 d e c i s i o n tree classification for finding cyber attacks with a high degree of accuracy and it used KDD99CUP dataset as input. Basically it detect whether this attacks are there or not like IPSWEEP, NEPTUNE, SMURF. Conclusions: Give brief concluding remarks on outcomes of what attacks are present and how to find.***

Keywords: Clustering, Classification, Decision trees, Feature, selection, Intrusion detection

1. Introduction

Now, A days various types of systems are available on internet .The popular used of internet has prompted network intrusion detection to become a critical component of infrastructure protection mechanisms. NIDS can be defined as identifying a set of malicious actions that threaten the integrity, confidentiality, and availability of a network resource. Intrusion detection is basically divided into two categories, i.e., anomaly detection and misuse detection. By using Misuse detection we can search for specific patterns or sequences of programs and user behaviors that match well-known intrusion scenarios. While, anomaly detection is used to develop models of normal network behaviors, and new intrusions are detected by evaluating significant deviations from the normal behavior. While accuracy is the essential requirement of an intrusion-detection system (IDS) its extensibility and adaptability are also critical in today's network computing environment An 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 large amount of audit records and the variety of system features, efficient and intelligent data analysis tools are required to discover the behavior of system activities. The increased reliance of government, military and commercial organizations on Internet technologies to conduct their everyday business creates a myriad of new challenges for cyber defense. The advancing complexity and variety of cyber attacks have almost rendered traditional IT defenses, such as anti-virus software or intrusion prevention systems, obsolete [1]. In addition, even the traditional strategy for defense that establishes multiple defensive layers so that penetration is next to impossible [2] is fast becoming increasingly ineffective as attackers continue to find and exploit vulnerabilities within systems. A cyber attack is a deliberate action against data, software or hardware that can destroy, degrade, disrupt or deny access to a networked computer system [3]. In recent years, data mining techniques have been employed with much success in the area of intrusion detection. In particular, the data preprocessing stage, which includes feature selection, has attracted much attention. Feature selection selects relevant subsets from the original dataset in order to minimize the effect of irrelevant and redundant features without greatly decreasing the accuracy of the classifier [8]. As a single IDS can be required to process a huge amount of data, the inclusion of erroneous, redundant or extraneous features in the analysis phase can make it difficult to detect anomalies amongst the noise increase the time and amount of processing power needed to analyze the data [9]. Therefore, an efficient feature selection technique should consider the relevancy of features, their discriminatory ability, and the number of features included in the analysis [11]. Although there are many candidate data mining techniques that can be employed, many of these techniques perform very well for detecting one specific cyber attack or specific set of cyber attacks under particular conditions.

2. Methodology

This paper presents a novel effects-based feature identification approach to identify sets of effects features that are statistically relevant which increased the accuracy of intrusion.

2.1. Data generation and collection

Acquiring data that is suitable for experimentation is a challenge within the area of intrusion detection. There are very few datasets available online, with the KDD evaluation dataset the most utilized in the existing body of research. Although this dataset includes five weeks of data and five categories of attack, there has been much debate as to the timeliness, complexity and simulation of a real network environment. In addition, as this research focuses in on the area where the actual effect takes place, this dataset was found to be lacking the necessary level of resolution needed to capture the effects of the attacks. Therefore, a cyber range was used to simulate a small office environment.

One of the fundamental challenges in the area of Cyber Defense and anomaly-based intrusion detection is identifying and defining normality. For the purpose of this research,

'normal' is defined as the everyday network traffic that is free of cyber attack. The dynamic nature of a network environment means that it is difficult to identify distinctive regions that separate all normal traffic from traffic that includes cyber attacks .To ensure that the tests in this experiment were valid, an additional dataset was collected that consisted of the 'normal' traffic without the attack scripts running.

The features used include: protocol, packet length, sequence number, time, duration, source IP, destination IP, source port, destination port, window size, MSS, IP header length, TCP flags, IP length, time to live, checksum, stream, designated services, expert message and frame protocol.

2.2. Data analysis

Having collected the four attack data sets on the cyber range, a data analysis design was developed involving three stages: (i) clustering, (ii) feature selection and (iii) decision tree classification. The first two stages result in subsets of effect features which are significantly relevant to a specific attack, with the third stage employing decision trees in order to validate the classification accuracy of these feature sets. These three stages are discussed below in more detail in the following subsections.

2.2.1. Stage 1: unsupervised clustering

As is the case with many modern day cyber attacks, the precise nature of the attack is generally not known before it is launched upon a system. As a result, the type and amount of 'training data' needed to train typical intrusion detection algorithms is near to impossible to obtain beforehand. In addition, part of the training process is to provide labeled data for which classification techniques can learn from. As intrusion detection is essentially a classification problem of 'attack' or 'not attack' in its base form, it is essential to include a phase that requires little or no training but is able to provide a labeled dataset. Unlike the majority of approaches to intrusion detection, this approach therefore employs clustering in the initial stage. Among a multitude of potential clustering algorithms, k-means was employed in the approach proposed in this paper for two reasons. First, k-means can be used to cluster together the 'normal' and the 'abnormal'. This clustering arrangement can then be used to label the data into attack/not attack for the subsequent stages described in the next section. Second, the clustering identifies features that are relevant to a particular attack; by analyzing the attributes of the data points in the abnormal clusters, the effect features that are most prevalent for a specific attack can be identified. k-means is ideal for such feature extraction because it provides transparent .

k-means is a partitional clustering algorithm developed by MacQueen. Its simplicity, efficiency and ease of implementation help to make it one of the most widely used clustering algorithms in use today. A description of the k-means algorithm is given below (adapted from).

Let X ={xi} i=1…..n be a set of n data points to be considered into a set of k clusters. K-means aims to find a natural grouping of clusters that minimizes the squared error between the empirical mean of a cluster and the points in the cluster. If μ_k is the mean of cluster c_k, the squared error between μ_k and the points in cluster c_k can be shown as: k-means is most typically used with the Euclidian distance metric, where the distance (d) between two points (x and y).

1. Randomly place the initial cluster centroids.
2. Assign each data point to its closest centroid
3. Recalculate centroid positions until cluster membership stabilizes.

Once the anomalous clusters have been identified and instances labeled, it was then possible to proceed.

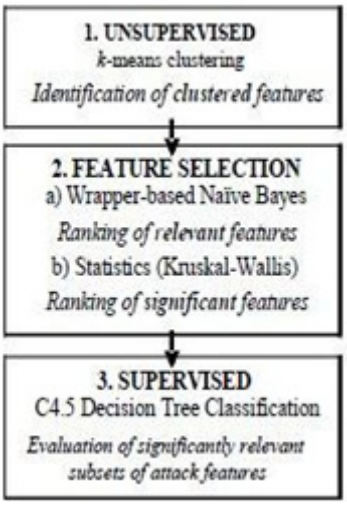

Figure 2: Effect Based Feature Identification data Analysis

2.2.2. Stage 2: feature selection

The second stage employs two different techniques with the aim of selecting both relevant and statistically significant feature sets that correspond directly to the effect of the attack. First, the Naïve Bayes classifier was used to identify and rank a relevant subset of features for each attack. Second, the Kruskal–Wallis statistical test was used to identify features that are statistically significant and rank them accordingly.

The Naïve Bayes classifier is widely used because of its ease of use, simplicity of implementation and interpretation, speed of training, ability to deal with missing values and because it does not require a large training sample . It was used in this research within wrapper-based feature selection to identify the features that were most relevant to each attack. Wrappers use classifiers to evaluate subsets and interactions of features and identify subsets of relevant features. The Naïve Bayes classifier was used within a greedy-stepwise wrapper in conjunction with 10 fold cross validation (10FCV). 10FCV is a commonly used method that divides the data into ten subsets, in which 9 of those subsets are considered as the training set with the remaining subset used to test. More specifically, features were given a score that reflected in how many folds the feature was highlighted as relevant. For example, if Feature 1 was highlighted in 8 out of the 10 folds, the score for that feature would be 80%. This score represents the relevancy of the feature in classifying instances into the class.

The Naïve Bayes classifier is a probabilistic approach to classification. It differs from Bayesian methods as it assumes

that variables are independent within each class, i.e. that the presence or(absence of)particular feature of class is unrelated to the presence (or absence)of any other feature.

Given an unclassified data point(X=x1…xn) the naïve bayes classifier will predict that x belongs to the class

2.2.3 Stage 3: decision tree classification

Decision trees are one of the most frequently used classification techniques and discover rules and relationships by systematically dividing information contained within data .Decision trees construct tree-like structures through a series of Boolean functions, i.e. "yes" or "no" questions based on the characteristics of a set of variables, until no more relevant branches can be derived. New data items can then be classified by starting at the root node and moving down through the branches until a leaf node is reached and a classification obtained. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier

Once the feature sets were identified using both types of feature selection described in the previous stage, they were then classified using the C4.5 decision tree algorithm. C4.5 is a descendant of ID3. C4.5 greatly improves upon ID3 as it can take both numeric and nominal attributes as inputs. It uses the divide and conquers approach to generate classifiers than can be expressed as decision trees or in the form of rule sets

Both the classification accuracies and the decision trees produced from the feature sets were then compared. The following section presents the results for the four attack datasets.

3. Mathematical Model

In order to formally reason and analyze an IDS, we firstly present a formal model of the IDS. Briefly, an IDS is represented as an eight-tuple (D*;* §*;* F*;* K*; S; R; P; C*), in which the first four items are data structures, and the last four are algorithms. Note that whenever we analyze and evaluate any IDS, we cannot talk about it without dealing with its data source. After all, our IDS model and framework are *data trace driven* and *evaluation oriented*.

D: the data source that an IDS will examine and analyze. Essentially this is a stream of consecutive data units. Since each IDS has its own unit of analysis, e.g., packet level or flow level for a network-based IDS (NISD), without loss of generality, we define D = (D_1*;* D_2*;* *:::*) where D_i is an analysis unit of data for the target IDS and D_i *2 fd*$_1$*;* d_2*;* *:::g*, d_j is the possible data unit. For example, an NIDS uses network traffic (packet stream), so the data source is a packet stream P = (P_1*;* P_2*;* *:::*). For a host-based IDS (HIDS) using system call sequence, the data source is a system call stream C = (C_1*;* C_2*;* *:::*). In this paper, we mainly take network data as our example, and packet as our data unit.

§: a finite set of data states indicating whether the data unit D_i is normal or anomalous (or further what type of intrusion). For convenience, we define an IDS oracle $Oracle_{IDS}$ which accepts any query with data unit D_i, and outputs an indication whether the unit is normal or anomalous. The IDS oracle knows the ground truth so it will always tell the truth[3]. Then for every data unit D_i, its state is $Oracle_{IDS}(D_i)$. The space of this state set is finite. For anomaly detection, § = *fNormal; Anomalousg*, or simply § = *fN; Ag*, or § = *f*0*;* 1*g* where 0 denotes normal and 1 denotes anomalous. For misuse detection, we can let § = *fNormal;* $AttackType_1$*;* $AttackType_2$*; :::g*, or § = *fN;* A_1*;* A_2*; :::g*.

F: a feature vector contains a finite number of features, formally F =< f_1*;* f_2*; :::;* f_n >. Every feature is a meaningful attribute of a data unit. For example, f_1 could be the protocol type (TCP, UDP, ICMP, etc.), f_2 could be the port number. Each feature has its own meaningful domain (called feature space) which is a set of discrete or continuous values (either Numerical or nominal). The full range of F is the product of the ranges of all the features. We denote it as *Range*(F) = f_1 £ f_2*:::* £ f_n.

K: The knowledge base about the profiles of normal/anomalous data. This knowledge base consists of profiling model (stored in some data structures) of normal and/or attack information. The detailed structure of K is possibly different for every IDS. It could be a tree, a Markov model, a Petri net, a rule set, a signature base, etc. For a signature-based NIDS, K is its rule set which contains only the attack profiling model (i.e., intrusion signatures). For an anomaly NIDS, K is mainly the profile of the normal traffic. Any activity that deviates the normal profile is considered as anomaly

S: feature selection algorithm. Given some D and the corresponding states $Oracle_{IDS}(D)$ (note sometimes only partial or even no such state information available), this algorithm should return several features f_i for the IDS to use. Although there is some preliminary effort to automatically generate worm signature [14, 22] for misuse IDSs as part of their features, generally speaking *S* still highly depends on domain knowledge and is normally conducted manually. The automatic selection or generation of features for both anomaly and misuse IDSs remains a grand challenge. The quality of features is one of the most important factors that will affect the effectiveness of an IDS.

R: data reduction and representation algorithm. When processing data, the IDS will firstly reduce the data and represent it in the feature space. This is a mapping/transition function, mapping the given data to a proper feature vector representation, namely *R* :D *!* F.

P: profiling algorithm, which is the procedure of generating the profiling knowledge base K. Given all the feature vector representations of data and their corresponding states, this algorithm will return the profiling knowledge base K.

C: classification algorithm. It is a mapping/transition function that maps the feature vector representation of given data to some states (it will also use the profiling base K in classification decision). Formally, *C* : F *!* §.

Most IDSs work in three steps.

1. Feature selection procedure: When we are developing an IDS, this is one of the first steps. Once the proper feature set is defined, it will be used in the following procedures. Normally, the feature selection procedure is conducted once, only during development.
2. Profiling procedure: sometimes also called training procedure [4]). We will run *P* (also involving *R*) on a sufficiently large amount of training data and get the profiling knowledge base K. Normally this procedure is performed once, only during development/training. In some situation, this procedure can be performed dynamically/periodically to update K.
3. Detection procedure: In this procedure, the IDS is used to detect intrusions in the data stream. This is the most important and frequently used procedure.

a) Original model (b) Abstract model *(c)* Clustered model

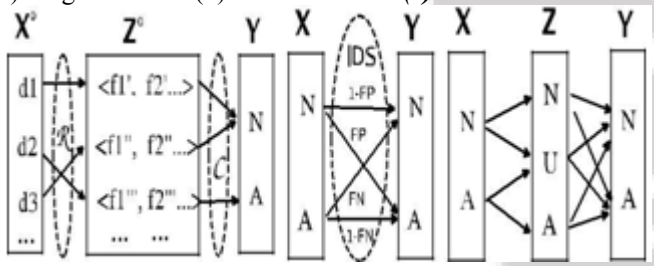

Figure 3: Intrusion Detection Procedure

The simple observation of this Markov chain data processing procedure motivates us to use information theory to analyze the process. Intuitively, we can roughly consider *R* as an encoding algorithm that uses feature vector to encode the original data unit. And then, *C*, as a decoding algorithm, decodes the feature representation to an output of the IDS. We should point out that although *R* and *C* resemble encoding and decoding procedures, they are not exactly the strict encoding and decoding schemes. In information theory, either encoding or decoding needs an encoding/decoding table containing all possible codewords for all possible source codes, so it can ensure a perfect encoding and decoding (without error or ambiguity). In the case of intrusion detection, we cannot enumerate all the possible input data units (source codes) and feature representations (code words), nor can we afford to store such a huge encoding/decoding table. As a result, both *R* and *C* algorithms can only work roughly correct, i.e., these algorithms may not guarantee errorless information transmission. We can analyze and quantify the effectiveness of this information transmission using information-theoretic metrics.

4. Implementation Details

4.1 Problem Statement and Solution Strategy

The increased reliance of government, military and commercial organizations on Internet technologies to conduct their everyday business creates a myriad of new challenges for cyber defense. The advancing complexity and variety of cyber attacks have almost rendered traditional IT defenses, such as anti-virus software or intrusion prevention systems, obsolete [1]. In addition, even the traditional strategy for defense that establishes multiple defensive layers so that penetration is next to impossible [2] is fast becoming increasingly in effective as attackers continue to find and exploit vulnerabilities within systems.

Detecting attacks that cross multiple applications/protocols or involve multiple phases is becoming extremely difficult with current methods. Moreover, scanning at a wider network level does not always provide the requisite resolution needed to determine the effect of the cyber attack, i.e. which features are effected in what manner at the level where the effect takes place.

4.2 Solution Strategy

The project proposes an effects-based method for hybrid intrusion detection that combines clustering, feature selection and classification in order to decrease the false alarm rate and pin point cyber attacks with a high degree of accuracy. Our solution involves three stages (i) clustering, (ii) feature selection and (iii) decision tree classification. The first two stages result in subsets of effect features which are significantly relevant to a specific attack, with the third stage employing decision trees in order to validate the classification accuracy of these feature sets.

4.3 Block Diagram

The system consists of following modules:

- Clustering: This module implements the K-means clustering on the input dataset & provides the clustered output to feature selection module
- Features Selection: This module implements the Naïve bayes algorithm to find the most relevant features.
- Decision tree classification: This module will implement the C45 algorithm to create the decision tree & provides to packet capture & analyze module.
- Packet Capture & Analyze : This module will capture live packets from the network & classifies to attack or not attack and also to type of attack

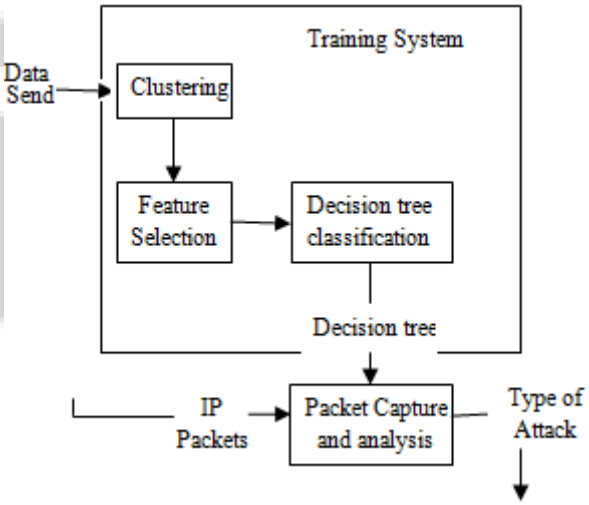

Figure 4: System Architecture

5. Conclusion

In network environments where the pace of operation is constantly moving and evolving, and the threat of cyber attack is always present, the need for reliable defenses is a crucial element in being able to identify, act and recover from unwanted network intrusions. Intrusion detection systems that rely solely on a database of stored known attacks are no longer sufficient for effectively detecting modern day attacks. This paper presents a novel anomaly detection technique that can be used to detect previously unknown attacks on a network by identifying the significantly relevant attack features. Awareness of these features will enhance the situational awareness of computer network operators and thus increase the effectiveness of computer network defense. The contributions of this work are as follows. First, the novel combination of techniques employed within this work, namely k-means clustering, Naïve Bayes, Kruskal–Wallis and C4.5, allows cyber attacks, as anomalies, to be pinpointed with a high degree of accuracy within the cluttered and conflicted cyber network environment. Further, the inclusion of the Naïve Bayes feature selection and the Kruskal–Wallis test in the approach facilitates the classification of both statistically significant and relevant feature sets, including a statistical benchmark for the validity of the approach. Unlike previous research in this area that does not focus on the specific features that characterize the cyber attack, this approach shows that a statistically relevant and reduced feature set filters out the noisy data associated with non-relevant features. In addition, this increases the efficiency of the analysis by speeding up and reducing the amount of processing required. These promising results encourage further research into the effect of a wider range of different classifiers and the exploration of recent developments within the clustering domain on the accuracy and performance of the technique; candidate methods for consideration include the evolutionary computational method and distributed Kalman filtering. Moreover, the power of the intrusion detection approach developed herein is its applicability for detecting different types of cyber attack and is a rich vein.

References

[1] J. Beale, J.C. Foster, J. Posluns, Snort 2.0 Intrusion Detection, Syngress Publish- ing, Inc, Rockland, 2003.
[2] S.X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: a review, Appl. Soft Comput. 10 (2) (2010) 1– 35.
[3] P. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, J. Network Comput. Appl.30 (1) (2007) 114–132.
[4] L. Zhou, F. Jiang, A rough set based decision tree algorithm and its application in intrusion detection pattern recognition and machine intelligence, Lect. Notes Comput. Sci. 6744/2011 (2011) 333–338.
[5] Urtubia, J.R. Perez-Correa, A. Soto, P. Pszczolkowski, Using data mining techniques to predict industrial wine problem fermentations, Food Control 18 (1) (2007) 1512–1517.
[6] J. Hair, W. Black, B. Babin, R. Anderson, R. Tatham, Multivariate Data Analysis, Prentice Hall, Upper Saddle River, NJ, 2006.
[7] T. Velmurugan, T. Santhanam, A survey of partition based clustering algorithms in data mining: an experimental approach, Inf. Technol. J. 10 (3) (2011) 478–484.
[8] J.C. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm,Comput. Geosci. 10 (2-3) (1984) 191–203.
[9] U. Premaratne, C. Ling, J. Samarabandu, T. Sidhu, Possibilistic decision trees for intrusion detection in IEC61850 automated substations, in: Proceedings of the 2009 International Conference on Industrial and Information Systems (ICIIS), Sri Lanka, December 2009, pp. 204–209.
[10] K. Badran, P. Rockett, Multi-class pattern classification using single, multi- dimensional feature-space feature extraction evolved by multi-objective genetic programming and its application to network intrusion detection, Genet. Program. Evol. Mach. 13 (1) (2012) 33–36.
[11] H.G. Kayacik, A.N.Z. Heywood, M.I. Heywood, Selecting features for intrusion detection: a feature relevance analysis on KDD99 intrusion detection data sets, in: Proceedings of the Third Annual Conference on Privacy, Security and Trust, Halifax, NS, Canada, October 2005.