IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 2, No. 2, June 2013, pp. 99~106
ISSN: 2252-8938

A Data Mining Approach for the Detection of Denial of Service Attack

Hoda Waguih
Departement of Computer and Information Systems, Sadat Academy for Management Sciences

Article Info

Article history:

Received Dec 08,2012
Revised Jan 05, 2013
Accepted March 02,2013

Keyword:

Data Mining
Machine Learning
Classification
Intrusion Detection
Denial of Service Attacks

ABSTRACT

Denial of Service (DoS) attacks constitutes one of the major threats and among the hardest security problems currently facing computer networks and particularly the Internet. A DoS attack can easily exhausts the computing and communication resources of its victim within a short period of time. Because of the seriousness of the problem many defense mechanisms have been proposed to fight these attacks. In this paper, we propose an approach that detects DoS attacks using data mining classification techniques. The approach is based on classifying "normal" traffic against "abnormal" traffic in the sense of DoS attacks. The paper investigates and evaluates the performance of J48 decision tree algorithm for the detection of DoS attacks and compares it with two rule based algorithms, namely OneR and Decision table. The selected algorithms were tested with benchmark 1998 DARPA Intrusion Detection data. Our research results show that both Decision tree and rule based classifiers deliver highly accurate results – greater than 99% accuracy – and exhibit high level of overall performance.

Corresponding Author:

Hoda Waguih,
Departement of Computer and Information Systems,
Sadat Academy for Management Sciences,
Cornish El Nile, Maadi, Cairo, Egypt.
Email: hoda.waguih@gmail.com

1. INTRODUCTION

The explosive growth of computer networks and particularly of the Internet has created many stability and security problems. One of the greatest threats that network security faces nowadays is Denial of Service attacks. The need for a defense against Denial of Service attacks is becoming an important challenge and a difficult task at the same time.

A Denial of Service DoS attack can be described as an attack designed to render a computer or network incapable of providing normal services [4]. A DoS attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user. These attacks don't necessarily damage data directly or permanently, but they intentionally compromise the availability of the resources.

The most common DoS attacks target the computer networks bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic that all available network resources are consumed and legitimate user requests cannot get through, resulting in degraded productivity. Connectivity attacks flood a computer with such a high volume of connection requests, that all available operating system resources are consumed and the computer can no longer process legitimate user requests.

In this paper, we will be presenting a comparative study of three data mining techniques for classification. The first of those is J48 pruned decision tree, a new version for the famous C4.5 decision tree classifier model by Quinlan [9]. The other two are rule based classifiers; namely OneR and Decision table. The main purpose of our experiment is to find the best performing classifier in terms of speed and accuracy

for the detection of denial of service attack on subset of the DARPA 1998 Intrusion Detection Evaluation Data, the KDDCup'99 data set.

There are many varieties of denial of service (or DoS) attacks in the KDD KDDCup'99 data set. Some DoS attacks (like a mailbomb, neptune, or smurf attack) abuse a perfectly legitimate feature. Others (teardrop, Ping of Death) create malformed packets that confuse the TCP/IP stack of the machine that is trying to reconstruct the packet. Still others (apache2, back, syslogd) take advantage of bugs in a particular network daemon. Table 1 provides an overview of the denial of service attacks used in our subset of the 1998 DARPA intrusion detection evaluation. Each row represents a single type of attack. The six columns show the attack name, a list of the services that the attack exploits, the platforms that are vulnerable to the attack, the type of mechanism that is exploited by the attack (implementation bug, abuse of feature, masquerading, or misconfiguration), a generalization of the amount of time the attack took to implement, and a summary of the effect of the attack [6]. The rest of the paper is organized as follows. Section 2 gives the basic principles of the three classifier used in our research study. Section 3 explains our conducted experiments on building a classifier model for the detection of denial of service attack using the audit data from the DARPA evaluation program. Section 4 provides a summary of the results obtained and their significance on the detection of denial of service attacks. Finally, Section 5 concludes the paper and provide insight to future work.

Table 1. Summary of Denial of Service Attacks

Attack	Service	Platform	Mechanism	Time	Effect
Back	http	Apach	Abuse/Bug	Short	Slow sever response
SYN flood	TCP	All	Abuse	Short	Deny service on one or more ports
Pod	Icmp	None	Bug	Short	None
Smurf	Icmp	All	Abuse	Moderate/long	Network slowdown
Teardrop	N/A	Linux	Bug	Short	Reboot machine

2. DATA MINING CLASSIFICATION METHODS

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [3]. This model is based on the analysis of a set of training data and may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks. In this section, we give an introduction to the classification techniques used for the detection of DoS attacks in the current research

2.1. Decision Tree Classifier

The decision tree classifier by Quinlan [9] is one of most well-known machine learning techniques. A decision tree is made of decision nodes and leaf nodes. Each decision node corresponds to a test X over a single attribute of the input data and has a number of branches, each of which handles an outcome of the test X. Each leaf node represents a class that is the result of decision for a case.

The process of constructing a decision tree is basically a divide and-conquer process [14]. Initially a set of examples representing the training data set consists of a number of classes. First, an attribute is selected to be placed at the root node and one branch is made for each possible value. This splits up the example set into subsets, one for every value of the attribute. The process can then be repeated recursively for each branch, using only those instances that actually reach the branch. If at any time all instances at a node have the same classification, we stop developing that part of the tree.

The problem here is how to choose the best attribute for each decision node during construction of the decision tree. The criterion that C4.5 chooses is Gain Ratio Criterion. The basic idea of this criterion is to choose an attribute which provides the maximum information gain at each splitting step while reducing the bias in favor of tests with many outcomes by normalization.

Once a decision tree is built, it can be used to classify testing data that has the same features as the training data. Starting from the root node of decision tree, the test is carried out on the same attribute of the testing case as the root node represents. The decision process takes the branch whose condition is satisfied by the value of tested attribute. This branch leads the decision process to a child of the root node. The same process is recursively executed until a leaf node is reached. The leaf node is associated with a class that is assigned to the test case. [10]

2.2. OneR Classifier

OneR, short for "One Rule", is a simple classification algorithm that generates a one-level decision tree expressed in the form of a set of rules that all test one particular attribute. OneR is a simple, cheap method that often comes up with quite good rules for characterizing the structure in data. Comprehensive studies of OneR's performance have shown it produces rules only slightly less accurate than state-of-the-art learning schemes while producing rules that are simple for humans to interpret. OneR is also able to handle missing values and numeric attributes showing adaptability despite simplicity

The OneR algorithm creates one rule for each attribute in the training data, then selects the rule with the smallest error rate as its 'one rule'. To create a rule for an attribute, the most frequent class for each attribute value must be determined. The most frequent class is simply the class that appears most often for that attribute value. A rule is simply a set of attribute values bound to their majority class; one such binding for each attribute value of the attribute the rule is based on. [14]

2.3. Decision Table Classifier

Decision tables provide an alternative way of representing knowledge in an understandable user-friendly way [13]. The origin of decision tables springs partly from the general use of tables to present information effectively and partly from the development of truth tables to define logic [8].

A Decision table is a tabular representation used to describe and analyze decision situations (e.g., classification of network traffic), where the state of a number of conditions jointly determines the execution of a set of actions [12]. In the context of rule based classifier, the conditions correspond to the antecedents of the rules; whereas the actions correspond to the outcome classes. The condition subjects are the criteria that are relevant to the classification or decision-making process. They represent the attributes of the rule antecedents about which information is needed to classify a given network traffic as normal or attack. The action subjects describe the possible outcomes of the decision-making process (i.e. the classes of the classification problem).

A Decision table is one type of classifier for which scheme-specific attribute selection is an essential part of the learning process. Therefore, the entire problem of learning decision tables consists of selecting the right attributes to include. Usually this is done by measuring the table's cross-validation performance for different subsets of attributes and choosing the best-performing subset. Fortunately, leave-one-out cross-validation is very cheap for this kind of classifier [14]. Obtaining the cross-validation error from a decision table derived from the training data is just a matter of manipulating the class counts associated with each of the table's entries, because the table's structure doesn't change when instances are added or deleted. The attribute space is generally searched by best-first search because this strategy is less likely to become stuck in a local maximum than others, such as forward selection

3. EXPERIMENTS

The dataset used in our experiments is a subset of the KDD Cup 99 dataset, which derives from the DARPA (Defense Advanced Research Projects Agency) dataset. In the 1998 DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks. Each record representing a TCP/IP connection composed of 41 features that are either quantitative or qualitative in nature [11].

A complete description of the features is available in [5], [7]. Instead of describing all the features here, we divided them into three groups [2]. The first group of attributes is the basic features of network connection, which include the duration, prototype, service, number of bytes from source IP addresses or from destination IP addresses, and some flags in TCP connections. The second group of attributes in KDD99 is composed of the content features of network connections and the third group is composed of the statistical features that are computed either by a time window or a window of certain kind of connections.

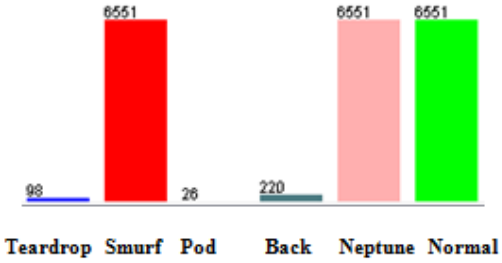

Figure 1. Class Distribution in the Data Set

In our experiments, we perform six-class classification representing normal traffic, and five denial of service attacks distributed as shown in figure 1. The (training and testing) data set contains a total of 19997 records, unevenly distributed across the Normal and the DoS attacks used. During examination of the data it turned out that the values of two nominal features (land, is_host_login) were constantly unchanged and five numeric features of the 41 features (urgent, num_failed_logins, su_attemp, num_shells, num_outbound_cmds) were zero over all data records. Obviously these features could not have any effect on classification and only made it more complicated and time consuming. They were, hence, excluded from the data vector. Hence the data vector was a 35 dimensional vector.

All experiments were performed in a computer with the configurations Intel(R) Core(TM) 2i5-2410 CPU 2.3GHz, 4 GB RAM, and the operation system platform is Microsoft Windows 7. The three algorithms used for classification are compared with the help of Weka.Weka is an open source data mining software that contains java implementations of many popular machine learning-algorithms including some popular classification algorithms. We selected three of the data mining classification algorithm, namely: J48 Pruned, OneR, and Decision table, for conducting our experiments. Though, prior to applying the selected classification algorithms, the target data set was transformed to the specific input data formats used by Weka.

The performance of a classification algorithm is determined by how accurately it classifies a given set of examples. In the conducted experiment, we tested the generated classifiers models using the k-fold Cross Validation mode The k-fold CV refers to a widely used experimental testing procedure where the dataset is randomly divided into k disjoint blocks of objects, then the data mining algorithm is trained using k-1 blocks and the remaining block is used to test the performance of the algorithm; this process is repeated k times. At the end, the recorded measures are averaged. In our experiments we choose k = 10 that is we used a 10 fold cross-validation for measuring the error rate of each classifier

3.1. J4.8 Decision Tree Classifier

The J48 Pruned decision tree is Weka implementation for the famous C4.5 decision tree classifier model by Quinlan [9]. In this experiment, we ran Weka J4.8 classifier to our dataset of 19997 instances, 35 attributes in order to differentiate between Normal and five types of DoS attacks, unevenly clustered. J4.8 is used with its default parameter settings with confidence threshold = 0.25 for pruning and the minimum number of instances permissible at a leave = 2 and a numfold = 3 determining the size of the pruning set. The resulting tree, with a size of 16 nodes and 9 leaves took 0.73 seconds to be built. To test and evaluate the resulting classifier, we applied a 10-fold cross-validation and received an accuracy of 99.97%

3.2. Rule-based Classifier

In this experiment, two rule based algorithm, OneR and Decision rule, implemented in weka are used on our data set. The OneR was selected based on the “simplicity first” methodology when analyzing practical data sets [14]. It generates a one-level decision tree expressed in the form of a set of rules that all test one attribute. In this experiment we ran WekaOneR classifier with the default parameters on our dataset of 19997 instances, 35 attributes. The classifier selected the Src_Byte as the minimum error attribute for prediction and generates 9 rules using this attribute. To test and evaluate the resulting classifier, we applied a 10-fold cross-validation and received an accuracy of 99.91%.

The other rule based algorithm in this experiment was the decision table. Weka builds a decision table majority classifier. It evaluates feature subsets using best-first search and can use cross-validation for evaluation. The Decision table classifier was used with its default parameter setting (crossVal = 1; searchMethod = BestFirst and useIBk = False) on our data set. A Decision table, as discussed before, is one type of classifier for which scheme-specific attribute selection is an essential part of the learning process. In

this experiment, a feature set of three attributes; Src_Byte, Dst_Host_Srv_Serror_Rate, and class was extracted, and 13 rules were generated using this feature set. To test and evaluate the resulting classifier, we applied a 10-fold cross-validation and received an accuracy of 99.985%

4. DISCUSSIONS AND RESULTS

To measure the effectiveness of our classifiers three metrics were used: *precision*, *recall*, and *overall accuracy*. These measures have been widely used in the data mining literature to evaluate data classification algorithms [14]. For a given class, the number of correctly classified objects is referred to as the True Positives. The number of objects falsely identified as a class is referred to as the False Positives. The number of objects from a class that are falsely labeled as another class is referred to as the False Negatives. Table 2

The *precision* of a classifier is the ratio of True Positives over the sum of True Positives and False Positives or the percentage of traffic flows that are properly attributed to a given class by this classifier.

$$\textbf{\textit{Precision}} = \frac{\textbf{TP}}{\textbf{\textit{TP + FP}}}$$

Recall is the ratio of True Positives over the sum of True Positives and False Negatives, or the percentage of traffic flows in class that are correctly identified

$$\textbf{\textit{Recall}} = \frac{\textbf{TP}}{\textbf{\textit{TP + FN}}}$$

Table 2. Standard metrics for evaluation of Attacks

Confusion Matrix (Standard metrics)		**Predicted Class**	
		Normal	**Attack**
Actual Class	**Normal**	**T**rue **N**egative	**F**alse **P**ositive
	Attack	**F**alse **N**egative	**T**rue **P**ositive

Overall accuracy is an aggregate precision and is defined as the sum of all True Positives to the sum of all the True and False Positives for all classes. This measures the overall accuracy of the classifier. Note that precision and recall are per-class measures.

$$\textbf{\textit{Overall Accuracy}} = \frac{\sum_{i=1}^{i=n} TPi}{\sum_{i=1}^{i=n} (TPi + FPi)}$$

where *n* is the number of classes.

Precision and recall are related to each other. If the Recall for one class is lower, this will cause the precision for other classes also to be lower because the algorithms used always classify the objects into a class. In addition, the overall accuracy is related to precision in that it measures the average precision of all the learned classes [1].

An initial analysis was performed to determine how well each attack is detected by each of the implemented classifiers. Three DoS attacks; Teardrop, Pod, and Neptune were fully detected by the three classifiers. One attack, Smurf, was fully detected by rule based classifiers. As for the remaining attack, back, 219 instance out of 220 were correctly classified by all the classifiers. For Normal traffic flows, 6549 out of 6551 were correctly classified by both the J48 and the J48 Decision tree classifiers as shown in table 3

The results for both Decision tree and rule based classifier models are shown in the tables 4-8. Table 4 through 6 depicts the confusion matrix for each of the learned classifiers. In our case we have six classes representing Normal and five DoS attack types. Therefore the confusion matrix is a 6 x 6 matrix. The number of correctly classified instances is the some of diagonals in the matrix; all other instances are incorrectly classified.

Table 3. Correctly Classified Instances Per DoS type

Name	Total Instances	Correctly Classified			Best Classifier
		J4.8	OneR	D. Table	
Tear	**98**	**98**	**98**	**98**	**All**
Smurf	6551	6548	6551	6551	Rule based
Pod	**26**	**26**	**26**	**26**	**All**
Back	220	219	219	219	All
Neptune	**6551**	**6551**	**6551**	**6551**	**All**
Normal	6551	6549	6534	6549	J4.8 & D.T

Table 4. Confusion Matrix for J4.8 Classifier

	Tear	Smurf	Pod	Back	Neptune	Normal
Tear	**98**	0	0	0	0	0
Smurf	0	**6548**	0	0	0	**3**
Pod	0	0	**26**	0	0	0
Back	0	0	0	**219**	0	**1**
Neptune	0	0	0	0	**6551**	0
Normal	0	0	0	**1**	**1**	**6549**

Table 5. Confusion Matrix for OneR Classifier

	Tear	Smurf	Pod	Back	Neptune	Normal
Tear	**98**	0	0	0	0	0
Smurf	0	**6551**	0	0	0	0
Pod	0	0	**26**	0	0	0
Back	0	0	0	**219**	0	**1**
Neptune	0	0	0	0	**6551**	0
Normal	0	1	1	**5**	**10**	**6534**

Table 6. Confusion Matrix for Decision Table Classifier

	Tear	Smurf	Pod	Back	Neptune	Normal
Tear	**98**	0	0	0	0	0
Smurf	0	**6551**	0	0	0	0
Pod	0	0	**26**	0	0	0
Back	0	0	0	**219**	0	**1**
Neptune	0	0	0	0	**6551**	0
Normal	0	**1**	**1**	0	0	**6549**

Table 7 shows the detailed accuracy per class among different classification methods. For every classifier we report the number of true positive (TP), the number of false positive (FP), Precision (P), and Recall (R) value for each DoS attack type and the Normal traffic instance. Table 8 shows the aggregate precision or the overall accuracy for each classifier. It is observed that both Decision tree represented by J48 and rule based classifier represented by OneR and Decision table showed highly accurate results – greater than 99% accuracy – and reveal high level of performance. The total accuracy for Decision table 99.985% outperforms both the Decision tree classifier 99.97% and the other rule based classifier OneR 99.91%. However, the time taken to build the OneR classifier is significantly shorter (0.36 seconds) as compared to the J48 Decision tree (1.36 seconds) and the Decision table (4.81 seconds) an advantage that is highly important in some situations where retraining needs to be done relatively quickly.

Table 7. Precision (P) and Recall (R) among different classification methods

	J4.8				OneR				Decision table			
Class	**TP**	**FP**	**P**	**R**	**TP**	**FP**	**P**	**R**	**TP**	**FP**	**P**	**R**
Tear drop	1	0	100	100	1	0	100	100	1	0	100	100
Smurf	1	0	100	100	1	0	100	100	1	0	100	100
Pod	1	0	100	100	**1**	0	**96.3**	100	1	0	**96.3**	100
Back	**99.5**	**0**	**99.5**	**99.5**	**99.5**	0	**97.8**	99.5	99.5	0	100	**99.5**
Neptune	1	0	100	100	**1**	0.1	**99.8**	100	1	0	100	100
Normal	**1**	0	**99.9**	100	99.7	**0**	100	**99.7**	**1**	**0**	100	100

Table 8. Accuracy among different classification methods

Classifier	Correctly		Incorrectly		Complexity
	#	%	#	%	
J4.8	19991	99.97	6	3	Tree size = 16, Leaves = 9 Time =1.39 s.
OneR	19979	99.91	18	9	Feature set = 1, 9 rules Time =0.36 s.
Decision Table	19994	99.985	3	1.5	Feature set = 3, 13 rules Time = 4.81 s.

5. CONCLUSIONS

Intrusion detection can be thought of as a classification problem: we wish to classify each audit record into one of a discrete set of possible categories, normal or a particular kind of intrusion. In our research, three classifiers; J48, OneR, and Decision table were developed to classify TCP/IP intrusion data to recognize whether a system is under denial of service attack. The classifiers are then evaluated and tested using 10-fold cross-validation and results are reported. It is observed that both Decision tree and rule based classifier deliver highly accurate results (> 99%) and exhibit high level of performance. The total accuracy for Decision table 99.985% outperforms both the Decision tree classifier 99.97% and the other rule based classifier OneR 99.91%. However, the time taken to build the OneR classifier is significantly shorter (0.36 seconds) as compared to the J48 Decision tree (1.36 seconds) and the Decision table (4.81 seconds) an advantage that is highly important in situations retraining needs to be done relatively quickly. Possible future development to the present study, involves comparing other types of classifiers, covering more attack scenarios in the dataset and addressing the fundamental issue of the unbalanced nature between normal and intrusive training examples for discriminative intrusion detection approaches.

REFERENCES

[1] Erman J, Mahanti A and Arlitt M. Internet traffic identification using machine learning. *In IEEE Globecom*. 2006.
[2] Farid D Md, Harbi N, Bahri E, Rahman MZ, and Rahman CM. Attacks Classification in Adaptive Intrusion Detection using Decision Tree. *World Academy of Science, Engineering and Technology*. 63: 2010.
[3] Han J and Kamber M. "*Data Mining: Concepts and Techniques*". Simon Fraser University, Morgan Kaufmann publishers, ISBN 1-55860- 489-8. 2001.
[4] Lee W and Stolfo S. *Data Mining Approaches for Intrusion Detection*. In Proceedings of the **Seventh USENIX Security Symposium (SECURITY '98),** San Antonio, TX. 1998.
[5] MIT Lincoln Laboratory, http://www.ll.mit.edu.
[6] Mukkamala S, Sung AH and Abraham A. "Intrusion detection using an ensemble of intelligent paradigms". *J. Network Computer Applications*. 2005; 28(2): 167-182.
[7] Mukkamala S. "*Intrusion detection using neural networks and support vector machine*". In Proceedings of the IEEE International Honolulu, HI. 2002.
[8] POOCH W. Translation of decision tables. *Computing Surveys 2*. 1974; 6.
[9] Quinlan JR. "*C4.5: Programs for machine learning*". Morgan Kaufmann, San Mateo, CA, 1993.
[10] Stein G, Chen B, Wu AS and Hua KA. "*Decision Tree Classifier for Network Intrusion Detection with GA-based Feature Selection*". in Proceedings of the 43rd ACM Southeast Conference, Kennesaw, GA. 2005.
[11] Stolfo S, Fan W, Lee W, Prodomidis A and Chan RK. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. *In DARPA Information Survivability Conference*. 2000.
[12] Vanthienen J, E Dries. Illustration of a decision table tool for specifying and implementing knowledge based systems. *Internat. J. Artificial Intelligence Tools*. 1994; 3(2): 267-288.
[13] Wets G, J Vanthienen, S Piramuthu. Extending a tabular knowledge based framework with feature selection. *Expert Systems Appl*. 1997; 13: 109-119.
[14] Witten IH and Frank E. "*Data Mining: Practical machine learning tools and techniques*". 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

BIOGRAPHY OF AUTHORS

Hoda Waguih is an Assistant Professor at the Department of Computer and Information Systems, Sadat Academy for Management Sciences, Cairo, Egypt. She got her Msc and Phdin Computer and Information Science from the Institute of Statistical Studied and Research (ISSR), Cairo University, Giza, Egypt. Herresearch interest includes diverse applications in artificial intelligence. She is particularly interested in machine learning, data mining and rough set theory applications in different domain areas.