Identifying False Alarm for Network Intrusion Detection System Using Data Mining and Decision Tree

NOR BADRUL ANUAR AND HASIMI SALLEHUDIN
Faculty *of* Computer Science and Information Technology
University *of* Malaya
50603 Kuala Lumpur
MALAYSIA

Abstract: -Although an intelligent intrusion and detection strategies are used to detect any false alarms within network critical segments of network infrastructures, reducing false positives are still being a major challenges. Up to this moment, these strategies focus on either detection or response features, but often lack of having both features together. Without considering those features together, intrusion detection systems are probably cannot highly detect on low false alarm rates. To offset abovementioned constraints, this paper proposes a strategy to focus on detection involving statistical analysis of both attack and normal traffics based on the training data of KDD Cup 99. This strategy is also included a hybrid statistical approach which using Data Mining and Decision Tree Classification. As a result, the statistical analysis can be manipulated to reduce misclassification of false positives and distinguish between attacks and false positives for the data of KDD Cup 99. Therefore, this strategy can be used to evaluate and enhance the capability of the IDS to detect and at the same time to respond to the threats and benign traffic in critical segments of network, application and database infrastructures.

Key-Word: - false positive, false negative, intrusion detection, data mining, decision tree.

1 Introduction

Within the period of Jun 2001 until November 2001, computer community around the world and also included Malaysia had been trapped with the biggest computer infrastructures attack in the Internet technology history. The statistical attacks reported by the Malaysian Computer Emergency Response Team (MyCERT) shows that 17,829 computers within that period had been infected by Nimda and Code Red attacks. The cost to recover all the damages by these attacks was estimated about RM22 million [5]. The amount was not including the cost for losing of the business opportunities due to the attacks. MyCERT argues that several precautions need to be taken in order to prevent viruses and other security threats infecting computers, which in turn can help to minimise the cost of recovery.

One possible precaution is the use of an Intrusion Detection System (IDS). IDS are an effective security technology, which can detect, prevent and possibly react to the attack [9]. It monitors target sources of activities, such as audit and network traffic data in computer or network systems, which deploys various techniques in order to provide security services. Therefore, the main objective of IDS is to detect all intrusions in an efficient manner [2]. For example, this may lead to an earlier detection of viruses and worms, and an early warning system in case of a computer virus outbreak. Besides, the effectiveness of IDS also needs to distinguish between incidents and "normal" alerts. This implies that while the number of false alarms should be reduced, real attacks should not go without noticed. Thus, it is important for IDS to be efficient so that the number of false positives and false negatives can be reduced [2]. In statistic, false positive and false negative are always referred as Type I error (i.e. also known as α error, or false positive) and type II error (i.e. also called as β error, or a false negative). These errors are normally used to describe possible errors made in a statistical decision process [6].

IDS also acts to labels alerts as incidents or as non-incidents. In an ideal situation, users may provide feedback by disagreeing or agreeing with the decision made by IDS. Normally, an input of IDS can be provided by one or more sensors. Multiple sensors can be used as input to a single analyser and works as observation points on the network [10]. These sensors normally generate a lot of alerts [13]. However, not all of these alerts are relevant. This is because all alerts are analysed, and only the relevant alerts are reported as incidents. The overview of this process is depicted in Figure 1. The input for this process, consisting of alerts, is provided by multiple sensors.

In order to have a real intrusion alarm, all activities need to be analysed by the analyser (see Figure 1). The efficiency of the result from the analysing process can be increased by using artificial intelligent and machine learning techniques. [8] mentions that some tasks cannot be easily defined properly by human expert compare to the effectiveness

of computer who is generating analysis. This is because human beings will find difficulty to find relationships and correlations in vast amounts of data [8]. During this research, an attempt is being made to filter incidents from alerts. The filtering process is needed to pre-identifying the real intrusion activities. This classification is done out by using data mining and decision tree techniques.

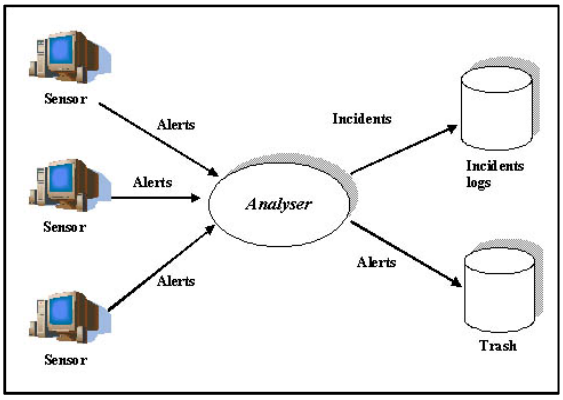


Fig.1: Process of alerts analysing that was generated by multiple sensors.

2 Intrusion Detection Systems Using Data Mining and Decision Tree

Detection method in IDS can be divided into two categories anomaly detection and misuse detection categories. Anomaly detection looks for something unusual and abnormal activities to a system or network, whereas misuse detection searches for something defined or pre-rules as bad manually by human. Anomaly detection systems regulate normal user behaviour profiles and also recognise intrusions by detecting some discrepancy from the normal behaviour. Although anomaly detection is occasionally able to detect previously unknown security attacks, it requires huge amount of data to be observed for producing user behaviour profiles. Besides, anomaly detection causes rather high false alarm rate because of any new user behaviour which is not including in user behaviour profile will be known as an intrusion [9].

Meanwhile, misuse detection can spots intrusion by matching security activities against predefined security attack patterns, which are stored in a database of previously known attacks. Misuse detection methods are usually able to recognise attacks with very high certainty, which is applied in a number of commercial IDS. However, misuse detection cannot identify novel intrusions, unless necessitates updating the database and the software system whenever new types of the security attacks are discovered [9].

The problem of identifying novel intrusion on misuse detection can be solved statistically in this project. We propose to use artificial intelligent technique such as data mining and decision tree. Data mining is the best options to be chosen because of previous well known attacks are stored in a database. A mining algorithm such as decision tree is used to analyse the previous known attack to generate a classifier for attacks. Accuracy of the algorithms is measured by the percentage of false positive and false negative that was generated during the classifying process. The higher of false positive means that the lower accuracy and precision of the classifier. Besides, the higher false negative implies that the recall of the classifier is lower.

Applying data mining with decision tree for the development of IDS provides some advantages compared to the classical approach. This is because decision tree gains more quantity of valuable information which in turn can help to enhance the decision on identifying the attacks. While IDS utilising crisp values may loose a large amount of valuable information but decision tree provides some flexibility to the uncertain problem of intrusion detection. This allows much greater complexity for IDS.

We perform experiments to classify the network traffic patterns according to the basic 5-class taxonomy, and also based on the 23-attack-instance taxonomy. The five classes of patterns in the DARPA data are discussed in the next section. According to [3], it is shown that using decision tree for classification gives high accuracy which in turn can help to reduce training time and testing time compare to the traditional neural network [3].

3 Intrusion Detection Data

In 1998, under DARPA intrusion detection evaluation programme, 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 [4]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [14]. Of this database a subset of 494021 data were used which compromised 20% of normal patterns. Attack types were divided into four main categories:

1. Probing: surveillance and other probing
2. DOS: denial of service
3. U2R: unauthorised access to local super user (root) privileges
4. R2L: unauthorised access from a remote machine

3.1 Probing

Probing is a class of attacks where an attacker scans a network to gather information in order to find known vulnerabilities. An attacker with a map of machines and services that are available on a network can manipulate the information to look for exploits. There are different types of probes: some of them abuse the computer's legitimate features; and some of them use

social engineering techniques. This class of attacks is the most commonly heard because it requires very little technical expertise.

3.2 Denial of service attacks

Denial of Service (DOS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, which in turn denying legitimate users access to a machine. There are different ways to launch DOS attacks: by abusing the computers legitimate features; by targeting the implementations bugs; or by exploiting the system's miss-configurations. DOS attacks are classified based on the services that an attacker renders unavailable to legitimate users.

3.3 User to root attacks

This attack is about an attacker starts out with access to a normal user account on the system by gaining root access. Regular programming mistakes and environment assumption give an attacker an opportunity to exploit the vulnerability of root access. Example of this class of attacks is regular buffer overflows.

3.4 Remote to user attacks

This attack is about an attacker sends packets to a machine over a network, then exploits machine's vulnerability to gain local access as a user illegally. There are different types of R2U attacks; the most common attack in this class is done by using social engineering.

The solution to classify this type of attack was done by many researchers with different approach and techniques. Most of them are using artificial intelligent approach such neural network, fuzzy logic, Bayesian, genetic algorithms and etc [3]. In this project, we apply decision tree technique based on C5.0 algorithms to classify the class of attack. The results of classification are presented by false positive and false negative numbers.

4 Result of Experiment

This project uses data mining software tools with the decision tree algorithms which known as See5/C5.0 version 2.04. The software is available for demo and evaluation which is provided by RuleQuest [11]. Since the software is depending on the capability of computer performance in order to build a classifier, installation must be made on a personal computer with Windows XP SP2 as operating system, 1.7GHz Pentium 4 processor and 512Mb of RAM.

The experiment was conducted in two (2) batches. The first experiment was using decision tree classifier onto the 10% of KDD Cup 99 training dataset. The second part of this experiment was using an algorithm of C5.0's rule-based classifier in order to compare the accuracy result with decision tree with same dataset.

4.1 Decision Tree

See5 was running so well for the decision tree classifier. Less than 2 minute was taken to train knowledge from 10% of dataset which contains about 494,021 of network traffics record. See5/C5.0 constructs a decision tree from the 494,021 training cases in the file of 'kddcup.data'.

The output of tree is very difficult to comprehend. Although it may not look much like a tree, this output can be paraphrased as *IF-THEN* statements.

The result showed that the C5.0 produced the tree with the non empty leaves with the size of 118 and the errors of the number and percentage of cases misclassified were 131 or (131/494021 * 100% = 0.03%) 0.03%.

This numbers of misclassified shows that C5.0 was the best option to use as IDS. The accuracy of C5.0 was measured in this experiment by the number of false positive and false negative produced. Table 1 showed the result based on the class of attacks and Table 2 showed the result based on the types of each attacks.

Table 1: The Total of False Positive and False Negative for the Class of Attack using Decision Tree

Class	Cases	False Positive	False Negative
Normal	97278	75	15
DoS	391458	9	19
Probe	4107	9	55
R2L	1117	6	23
U2R	59	32	19
Total	**494021**	**131**	**131**

The result showed that only 75 cases were false positive with the ratio of 0.08% (75/97278 * 100% = 0.08%) or 99.92% (100% – 0.08% = 99.92%) statistical significant accurate. The number of 75 cases implied that the failure of the classifier to classify the attacks as a normal. While, only 15 cases of normal record were classified as attacks.

The C5.0 was significantly accurate to classify the DoS attacks with the ratio of 99.99%, Probe is 99.78%, and R2L is 99.46%. However, accuracy ratio for U2R attack was very low with 45.76% respectively not accurate. This implied that, data mining and decision tree using C5.0 algorithms were not suitable to classify the U2R attack because of the number of the record is very small in training dataset.

Table 2 showed the detail of the total of false positive and false negative for types of attack in

decision tree. The totals of 131 attacks were classified as false positive and negative.

According to the number of false positive and false negative is 0, the experiment was significantly accurate to classify the phf and teardrop attacks. Smurf attack has a biggest number of the record and C5.0 significantly able to classify 99.99% of the record as a Smurf attack. Phf is the second smallest number in the record and significantly successfully classified as phf. However, Spy attack is the smallest number of the record and significantly misclassified as different types of attack record. These mean that, even the number of cases is smallest or biggest; classification is not depending on it but it is depending on the value of 41 attributes representing. The values of the attributes are very similar to each other especially normal record which is misclassified as satan (12), warezclient (10), back (9) and etc. (see Appendix A).

Table 2: The Total of False Positive and False Negative for Types of Attack Decision Tree

Attacks	Cases	False Positive	False Negative
back	2203	1	9
buffer_overflow	30	4	2
ftp_write	8	0	5
guess_passwd	53	0	2
imap	12	0	2
ipsweep	1247	3	3
land	21	1	0
loadmodule	9	0	7
multihop	7	1	3
neptune	107201	28	1
nmap	231	1	10
normal	97278	75	15
perl	3	1	0
phf	4	0	0
pod	264	0	5
portsweep	1040	1	27
rootkit	10	0	9
satan	1589	4	15
smurf	280790	6	2
spy	2	0	2
teardrop	979	0	0
warezclient	1020	3	10
warezmaster	20	2	2
Total	**494021**	**131**	**131**

From the 41 attributes of record KDD Cup '99, only 20 of the attributes were used for C5.0 decision trees classifier. The ratios of the attribute usage are shown in the Table 3. Some machine learning technique such as neural network, fuzzy logic and support vector machine (SVM) are depending on the input attributes. So, these attribute usage can apply to neural network, fuzzy logic and SVM without using all the 41 attributes to run the classification but only using the most usage attribute for classification [1].

Table 3: Attribute usage for decision tree classifier

Attributes	Percentage
wrong_fragment	100%
Land	100%
same_srv_rate	100%
dst_host_diff_srv_rate	99%
src_bytes	83%
dst_host_serror_rate	78%
num_compromised	78%
num_failed_logins	77%
dst_host_srv_diff_host_rate	77%
Hot	77%
root_shell	77%
dst_host_same_src_port_rate	77%
Duration	77%
srv_serror_rate	77%
protocol_type	57%
dst_host_srv_count	19%
dst_bytes	18%
Count	18%
logged_in	2%
dst_host_srv_rerror_rate	1%

4.2 Rule-based Classifiers

Decision trees can sometimes be quite difficult to comprehend when the size of tree is too big. To offset the drawback of decision trees, See5 can generate classifiers called *rulesets* that consist of unordered collections of (relatively) simple if-then rules. Therefore, the second experiment is applying the rule-based algorithms to classify the KDD dataset. As similar with decision tree, time used for the classification process is still less than 2 minutes.

Rulesets are generally easier to understand compare to the trees since each rule describes a specific context associated with a class or an attribute. Furthermore, a ruleset are more comprehensibility than tree. From Table 4, the total of 72 ruleset was constructed with 187 (0.04%) misclassification errors. Table 4 showed the split number of rules for each types of attack. There was 23 ruleset which was constructed to classify the normal record. Three attack types are not constructed the ruleset by C5.0. There are phf, spy and warezmaster attacks. These three attacks are 100% confident classified as false negative. Any record that representing by phf, spy and warezmaster will be classified as normal or other attacks because there is no rules were generated to classify this attack as phf, spy or warezmaster. The miss-generated rules for these attacks are because of small number of record representing by them or their attribute is similar to other types of attack or normal.

The result in Table 5 showed that 128 number of attack record were classified as normal (false positive). In this case, Table 6 show the C5.0 accuracy in order to classify the DoS attacks was 99.99%, Probe was 99.85%, and R2L was 99.82% and 94.92% respectively. The total of false positive and false negative for the class of attack using rule-based classifier is shown in Table 6.

Table 4: Split Number of Ruleset

Attacks	Cases	Number of Rules
back	2203	**1**
buffer_overflow	30	**3**
ftp_write	8	**1**
guess_passwd	53	**1**
imap	12	**1**
ipsweep	1247	**3**
land	21	**1**
loadmodule	9	**1**
multihop	7	**2**
neptune	107201	**3**
nmap	231	**2**
normal	97278	**23**
perl	3	**1**
phf	4	**0**
pod	264	**1**
portsweep	1040	**7**
rootkit	10	**0**
satan	1589	**7**
smurf	280790	**3**
spy	2	**0**
teardrop	979	**1**
warezclient	1020	**9**
warezmaster	20	**0**
	494021	**72**

Table 5: The Total of False Positive and False Negative for Rule-based classifier

Attacks	Cases	False Positive	False Negative
back	2203	0	9
buffer_overflow	30	1	8
ftp_write	8	0	6
guess_passwd	53	0	1
imap	12	0	2
ipsweep	1247	2	3
land	21	1	0
loadmodule	9	0	7
multihop	7	1	3
neptune	107201	26	4
nmap	231	0	27
normal	97278	128	12
perl	3	1	0
phf	4	0	4
pod	264	0	5
portsweep	1040	2	24
rootkit	10	0	10
satan	1589	2	20
smurf	280790	21	2
spy	2	0	2
teardrop	979	0	0
warezclient	1020	2	18
warezmaster	20	0	20
	494021	**187**	**187**

Table 6: The Total of False Positive and False Negative for the Class of Attack using Rule-based Classifier

Class	Cases	False Positive	False Negative
Normal	97278	128	12
DoS	391458	48	20
Probe	4107	6	74
R2L	1117	2	49
U2R	59	3	28
	494021	**187**	**187**

The number of observations and conclusion are drawn from the results illustrated in Table 7. Result of Table 7 showed that performances of the comparison of false alarm rate show that decision tree was more accurate to classify class of Normal, DoS and R2L than rule-based classifier. However, rule-based classifier was more accurate to classify class Probe and U2R than decision tree because of rule-based classifier generate lower false alarm rate that decision tree classifier.

Table 7: Performance Comparison of False Alarm Rate for the Class of Attack using Decision tree and Rule-based Classifier

Class	False Alarm Rate for Decision Tree (%)	False Alarm Rate for Rule-based Classifier (%)
Normal	0.015	0.025
DoS	1.822×10^{-3}	9.716×10^{-3}
Probe	1.822×10^{-3}	1.215×10^{-3}
R2L	1.215×10^{-3}	4.048×10^{-4}
U2R	6.477×10^{-3}	6.073×10^{-4}

5 Conclusion

The accuracy of decision tree to classify the normal record is higher than rule-based classification (see Table 7). Since the acceptable levels of false alarms for IDS is less then 10%, decision tree classification and rule-based classification is suitable to use as IDS model because of the false alarm rate for class normal is 1.5% and 2.5% in Table 7. However, the acceptable levels of false alarm can be higher or lower depending on the level of IDS tuning and the type of traffic on a network [12]. In this project, we have proved the importance of decision tree for modeling intrusion detection for class of normal, DoS, and R2L. While for the class of Probe and U2R, rule-based classification is suitable to apply. However, based on acceptable levels of false alarm rate, decision tree is more suitable than rule-based for modeling intrusion detection systems.

References:

[1] Ajith Abraham, Ravi. Jain**,** *Soft Computing Models for Network Intrusion Detection Systems*. Classification and Clustering for Knowledge Discovery, Saman Halgamuge and Lipo Wang (Eds.), Studies in Computational Intelligence, Vol. 4, Springer Verlag Germany, 2005, ISBN: 3-540-26073-0, Chapter 13, pp. 187-204.

[2] Gowadia, V., Farkas, C., and Valtorta, M., Paid: A probabilistic agent-based intrusion detection system. *Journal of Computers and Security,* 2005

[3] Hettich, S. and Bay, S. D.,*The UCI KDD Archive* Irvine, CA: University of California, Irvine, KDD Cup 1999 Data, 5th International Conference on Knowledge Discovery and Data Mining, 1999.

[4] Kendall, K. 1999, *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*, S.M. Thesis, MIT Department of Electrical Engineering and Computer Science, 1999.

[5] Malaysian Computer Emergency Response Team (MyCERT), 2007 http://www.mycert.org.my [Last visit: 05-12-2007].

[6] Moulton, R.T., "Network Security", *Datamation*, Vol.29, No.7, 1983, pp.121-127

[7] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation. [Internet] http://www.ll.mit.edu/IST/ideval [Last visit: 06-12-2007].

[8] Nilsson, N., *Introduction to Machine Learning*. Stanford University, 1996. [Internet] http://ai.stanford.edu/~nilsson/MLDraftBook/MLBOOK.pdf.[Last visit: 05-12-2007].

[9] Rebecca Base and Peter Mell, *NIST Special Publication on Intrusion Detection Systems*. Infidel, Inc., Scotts Valley, CA and National Institute of Standards and Technology, 2001.

[10]Rietta, F., Application layer intrusion detection for sql injection. *Proceedings of the 2006 ACM Symposium of Applied Computing (ACMSE-2006)*,

[11]RuleQuest, (2007) [Internet] http://www.RuleQuest.com [Last visit 06-12-2007]

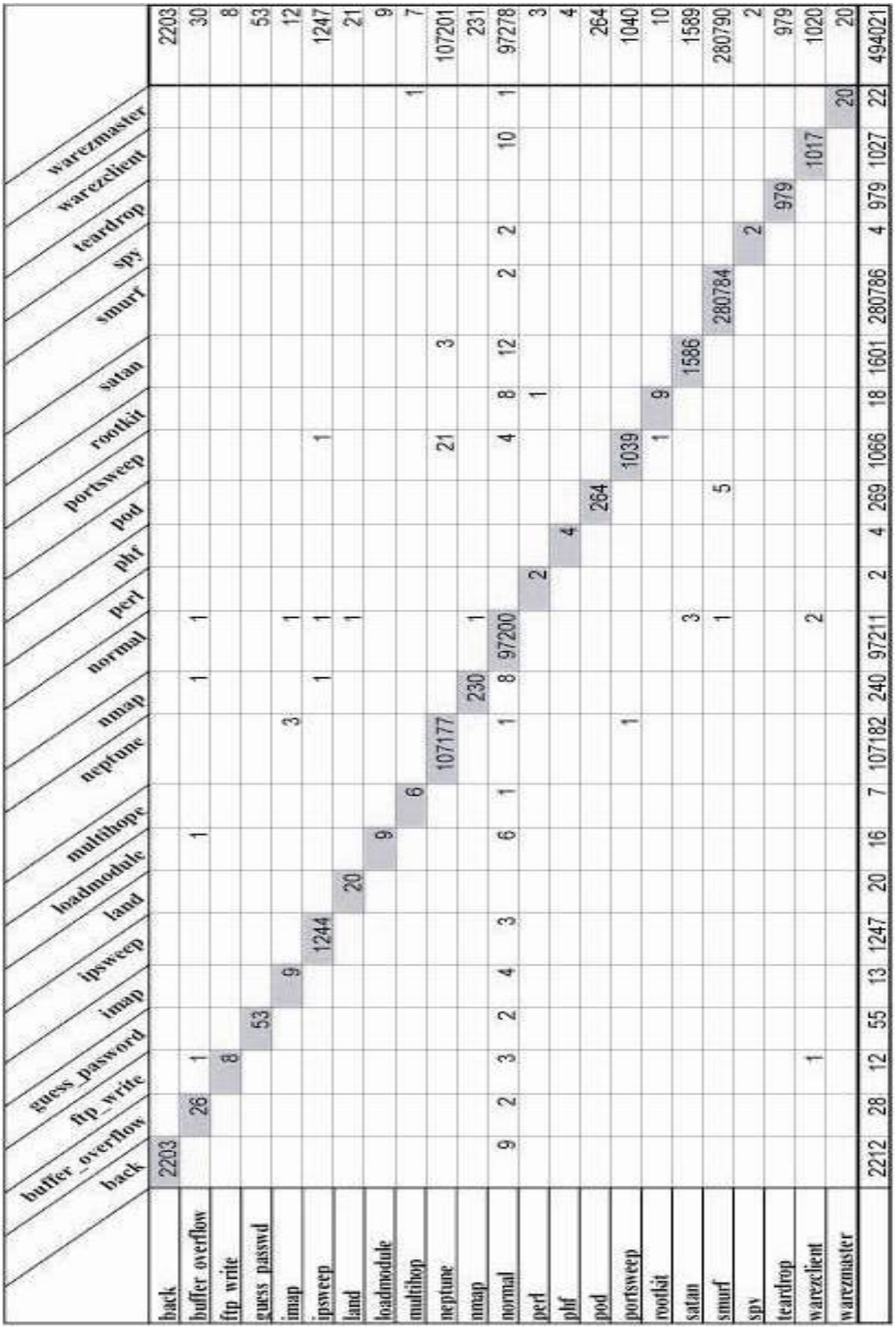
[12]Securityfocus, (2007). [Internet] http://www.securityfocus.com/infocus/1463 [Last visit: 16-1-2008]

[13]Varine, B., *Intrusion Detection FAQ: Should we outsource monitoring?* SANS Institute, 2001 [Internet] http://www.sans.org/resources/idfaq/outsource.php .[Last visit: 05-12-2007].

[14]Wenke Lee, Sal Stolfo and Kui Mok, *A Data Mining Framework for Building Intrusion Detection Models*. Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, CA.

APPENDIX A:

Confusion Matrix for Decision Tree Classification



	back	buffer_overflow	ftp_write	guess_passwd	imap	ipsweep	land	loadmodule	multihop	neptune	nmap	normal	perl	phf	pod	portsweep	rootkit	satan	smurf	spy	teardrop	warezclient	warezmaster	
back	2203																							2203
buffer_overflow		26																						30
ftp_write			8																					8
guess_passwd				53																				53
imap					9						3		1											12
ipsweep						1244						1	1				1							1247
land							20						1											21
loadmodule								9																9
multihop									6														1	7
neptune										107177							21		3					107201
nmap											230		1											231
normal	9	2	3	2	4	3		6	1	1	8	97200					4	8	12		2	10	1	97278
perl													2					1						3
phf														4										4
pod															264									264
portsweep										1						1039								1040
rootkit																	1	9						10
satan												3							1586					1589
smurf												1				5			280784					280790
spy																				2				2
teardrop																					979			979
warezclient			1									2										1017		1020
warezmaster																							20	20
	2212	28	12	55	13	1247	20	16	7	107182	240	97211	2	4	269	1066	18	1601	280786	4	979	1027	22	494021