Detecting Known and Novel Network Intrusions

Yacine Bouzida[1]* and Frédéric Cuppens[2]

[1] Mitsubishi Electric ITE-TCL 1, allée de Beaulieu CS 10806 F-35708, Rennes
Bouzida@tcl.ite.mee.com
[2] ENST Bretagne 2, rue de la Châtaigneraie F-35576, Cesson Sévigné
Frederic.Cuppens@enst-bretagne.fr

Abstract. It is well known that signature based intrusion detection systems are only able to detect known attacks. Unfortunately, current anomaly based intrusion detection systems are also unable to detect all kinds of new attacks because they are designed to restricted applications on limited environment. Current hackers are using new attacks where neither access control systems nor current signature based systems can prevent the devastating results of these attacks against information systems. We enhance the notion of anomaly detection, introduce necessary conditions that should be taken into account by the building detection models and propose a new machine learning algorithm based on decision trees to discover known and unknown attacks in real time. Experimental results demonstrate that the proposed method is highly successful in detecting new attacks and significantly outperforms previous work.

1 Introduction

Anomaly intrusion detection systems are not as well studied or explored as misuse detection ones. Misuse detection consists in using patterns of well known intrusions to match and identify known labels for unlabeled data sets. In fact, many commercial and open source intrusion detection systems are misuse based ones. Recently, attackers have explored serious break-ins to many commercial and government sites where serious damages have occurred. The different intrusions that have been used were new. This situation was foreseeable because the attackers are attempting to develop new attacks forms where neither misuse detection tools nor access control tools installed in our networks may detect or stop these new attacks forms.

By contrast, anomaly detection consists in building profiles of normal behaviors then detecting any deviation of a new behavior from the learned normal profiles. This definition of anomaly detection is restrictive because only one class which corresponds to the normal behavior is learned.

In this paper, we extend the definition of anomaly detection to not only take into account normal profiles but also handle known attacks and explore supervised machine learning techniques, particularly decision trees. These techniques have proven their efficiency in predicting the different classes of the unlabeled

* This work was completed when the author was a PhD student at ENST Bretagne.

data in the test data set for the KDD99 intrusion detection contest. Since machine learning techniques, generally, cannot find boundaries between known and unknown classes, an extension of decision trees is introduced to deal with new unknown anomalies.

The rest of the paper is organized as the following. Section 2 presents our motivations for extending the notion of anomaly detection. Section 3 enhances machine learning techniques, particularly decision trees, to handle new instances that are not considered in all current supervised machine learning techniques. Using the improvement of decision trees suggested in Section 3, Section 4 describes the experimental results obtained, using the decision trees algorithm and the modified algorithm, over the DARPA98 intrusion detection data set [3]. The KDD99 intrusion detection contest [5] uses a version of this data set. The data set provided in DARPA98 has been severely criticized in several previous works. However, we explain why this data set remains interesting to experiment our proposal. The first results obtained with our enhanced algorithm over KDD99 do not correspond to what we expect. This is due, in reality, to the transformation of DARPA98 to KDD99. Section 5 explains why KDD99 is not an appropriate transformation of DARPA99 and suggests necessary conditions a transformation technique should satisfy in order to keep maximum data information while transforming *tcpdump* traffic into connection records. Section 6 presents the results we obtained when considering new attacks not present in DARPA98 and Section 7 offers conclusive remarks and discusses future work.

2 Motivations

Anomaly intrusion detection is the first intrusion detection method that was introduced to monitor computer systems by Anderson [1] in 1980. At that time, intrusion detection was immature since only user behavior and some system events were taken into account. In fact, this approach consisted in establishing normal behavior profile for user and system activity and observing significant deviations of the actual user activity with respect to the established habitual profile. Significant deviations are flagged as anomalous and should raise suspicion. This definition did not take into account the expert knowledge of known vulnerabilities and then known attacks. This is why we enhance the notion of anomaly detection not only by considering normal profiles but also by taking into account abnormal behaviors that are extracted from known attacks.

Since we have knowledge about known vulnerabilities and their corresponding attacks, we may enhance the anomaly detection by adding to the learning step the abnormal behavior corresponding to known attacks. Therefore anomaly detection would consists in learning all known *normal* and *attacks* profiles. Based on this knowledge, anomaly detection has then to detect whether a new observed profile is normal or abnormal and its corresponding known attack is determined or the observed profile is new and therefore is considered as a novel unknown behavior. Thereafter, we suggest that a diagnosis should be done on the observed traffic that has caused the detection of the new anomaly in order to find out the

reason of this new observation. If it corresponds to a normal new activity that was never observed before it is flagged as a normal profile or as a new attack. The new observations with their real classification would then be considered for further investigation. We note that the diagnosis of new observed behaviors is not an objective of this paper. This will be discussed in a forthcoming paper.

Current supervised machine learning and classification techniques are not written to detect new classes that are not present in the training data set (new profiles that are not seen before in our case). Therefore, we investigate in the following section the decision trees induction algorithm and improve it in order to deal with these new classes. We choose to use decision trees induction algorithm to best clarify the idea of new cases since it is such an illustrative technique. In addition, it is the best winning entry [4] for KDD99 intrusion detection contest. Added to this the fact that we are familiar with this technique since we used it to detect intrusions by combining it with principal component analysis for space and time reduction [2].

3 Decision Trees Enhancement

Decision trees classifiers are based on the *"divide and conquer"* strategy to construct an appropriate tree from a given learning set S containing a finite and not empty set of labeled instances. In the following, we are interested in the C4.5 Quinlan algorithm [7].

Most of the decision trees algorithms use a top down strategy; i.e. from the root to the leaves. Two main processes are necessary to use the decision tree, respectively called the building process and classification process.

3.1 Building process

It consists in building the tree by using the labeled training data set. An attribute is selected for each node based on how it is more informative than others. Leaves are also assigned to their corresponding class during this process.

To measure how informative a node is, Shanon entropy is used to construct the decision trees. The selection of the best attribute node is based on the gain $Gain(S, A)$ where S is a set of records and A a non categorical attribute. This gain defines the expected reduction in entropy due to sorting on A. It is calculated as the following [7]:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{\mid S_v \mid}{\mid S \mid} Entropy(S_v) \tag{1}$$

If we consider only $Gain(S, A)$ then an attribute with many values will be automatically selected.

One solution is to use $GainRatio$ instead [7]

$$GainRatio(S, A) = \frac{Gain(S, A)}{-\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}} \tag{2}$$

where S_i is a subset of S for which A has a value v_i.

3.2 Classification process

A decision tree is important not because it summarizes what we know, i.e. the training set, but because we hope it will classify correctly new cases. Thus, when building classification models, one should have both training data to build the model and test data to verify how well it actually works. New instances are classified by traversing the tree from the up to down based on their attribute values and the node values until one leaf is reached that corresponds to the class of the new instance.

3.3 Improving the classification process

The decision trees C4.5 algorithm written by Quinlan presents a drawback toward the set of instances that are not covered by any of the rules generated from the decision tree. He proposes a default class for those instances. The default class is defined as the one with most items not covered by any rule. In the case of conflict, ties are resolved in favor of more frequent classes.

Using this principle, a default class from the learning data set is assigned to any observed instance that may be normal, known or unknown attack.

The default class is assigned to any new instance which is not covered by any rule generated from the training data set. This classification is useful only in the case of an exclusive classification; i.e. there is a class for any given instance and the assigned class has at least one instance in the learning data set. Since we are interested in detecting novel attacks this classification kind would not be able to detect new attacks that normally are not covered by any rule from the tree built during the learning step.

To resolve this problem, we introduce the following principle: A default class denoted *new class* is assigned to any new class that does not have a corresponding class in the training data set. Therefore, if any new instance does not match any of the rules generated by the decision tree then this instance is classified as a new class instead of assigning it to a default class.

To illustrate the effectiveness of this new idea, in Section 4, we conduct our experiments on the KDD99 database since it contains many new attacks in the test data set that are not present in the training data set and on real traffic in our laboratory network where some new attacks that were not available when DARPA98 was built such as the slammer worm and the different DDoS attacks are presented in Section 6.

4 Experimental Analysis of KDD99

The main task for the KDD99 classifier learning contest was to provide a predictive model able to distinguish between legitimate and illegitimate connections in a computer network. The training data set contained about $5,000,000$ connection records, and the training 10% data set consisted of $494,021$ records among which there are $97,278$ normal connections (i.e. 19.69%). Each connection record

consists of 41 different attributes that describe the different features of the corresponding connection, and the value of the connection is labeled either as an attack with one specific attack type, or as normal. There are 39 different attack types present in the 10% data sets. We notice that there are many attacks that are present in the test data set but do not have any occurrence in the learning data set such as *saint*, *mailbomb*, *httptunnel*, *snmpguess*, etc.

Each attack type falls exactly into one of the following four categories: *probing, DoS, U2R* and *R2L*.

The task was to predict the type of each connection in the test data set containing 311, 029 connections.

There are many occurrences of new attack forms for the two classes U2R and R2L in the test data set. The Probing class presents also many occurrences of new attacks forms in the test data set. However, for this class the difference is in the name of the tool used for the scan operation, not in the method with which the probing is performed.

We should mention that the different attacks present in the test data set that do not have any occurrence in the training data set cannot be easily classified into their appropriate class and will be classified in the class that has a form close to theirs and generally to the normal class. However, if the connection form does not characterize precisely the corresponding attack or the normal traffic as its initial *tcpdump* form then the classification of the new attacks would be unforeseeable.

To rank the different results a cost matrix C is defined. A cost per test (CPT) was calculated using the formula given in Equation 3.

$$CPT = \frac{1}{N} \sum_{i=1}^{5} \sum_{j=1}^{5} C_{i,j} * CM_{i,j} \tag{3}$$

where C corresponds to the cost matrix, N is the number of instances in the test data set and CM corresponds to the confusion matrix.

In the following, we present the different experiments and results obtained when using the different rules generated from the standard C4.5 algorithm. In the second step, the enhanced C4.5 algorithm, as explained in Section 3 to handle new instances, is used.

The accuracy of each experiment is based on the cost per test and the percentage of successful prediction (PSP) on the test data set.

$$PSP = \frac{number\ of\ successful\ instance\ classification}{number\ of\ instances\ in\ the\ test\ set} \tag{4}$$

Table 1 presents the confusion matrix for the 5 classes when using the rules from the decision trees generated by the standard C4.5 algorithm of Quinlan [7].

From Table 1, the two classes R2L and U2R are badly predicted. On the other hand, many probing and DoS instances are misclassified within the normal class. Most misclassified instances are predicted as normal. This is due to the supervised C4.5 algorithm that assigns a default class among known classes as explained in Section 3.

Predicted Actual	%Normal	%Probing	%DoS	%U2R	%R2L
Normal(60,593)	**99.47**	0.40	0.12	0.01	0.00
Probing (4,166)	18.24	**72.73**	2.45	0.00	6.58
DoS (229,853)	2.62	0.06	**97.14**	0.00	0.18
U2R (228)	82.89	4.39	0.44	**7.02**	5.26
R2L (16.189)	81.60	14.85	0.00	0.70	**2.85**
$PSP = 92.30\%$, $CPT = 0.23425$					

Table 1. Confusion Matrix relative to the five classes using the rules generated by the standard C4.5.

Hence, if a new instance is presented (different from all other known normal or abnormal instances), it is automatically classified as the default class *normal* since it has the highest number of uncovered instances.

Table 2 shows the confusion matrix obtained when using the enhanced C4.5 algorithm that we have modified to affect a class labeled *new* to any uncovered or unseen instance.

Predicted Actual	%Normal	%Probing	%DoS	%U2R	%R2L	**%New**
Normal(60,593)	**99.43**	0.40	0.12	0.01	0.00	**0.04**
Probing (4,166)	8.19	**72.73**	2.45	0.00	6,58	**10.06**
DoS (229,853)	2.26	0.06	**97.14**	0.00	0.18	**0.36**
U2R (228)	21.93	4.39	0.44	**7.02**	5.26	**60.96**
R2L (16,189)	79.41	14.85	0.00	0.70	**2.85**	**2.20**
$PSP = (92.30 + (0.57))\%$, $CPT = 0.2228$						

Table 2. Confusion matrix when using the generated rules from the enhanced C4.5 algorithm.

Using the new enhanced C4.5 algorithm, we have increased the detection rate of the $U2R$ class by 60.96% which decreases the false negative rate of this class from 82.89%(189/228) to $21,93\%(50/228)$. The detection rate of the Probing class is also enhanced by $10,06\%$ corresponding to 413 instances which are not classified as a normal traffic but as a new class, hence as a new attack.

We should mention that the highest ratio for the U2R class has never exceeded 14% according to the different results available in the literature. Using our approach this attack is detected as an abnormal traffic with a detection rate of 67.98%.

However, the false negative rate of the R2L class remains stable. In addition, even if we count the detection ratio of the new instances that are classified as new attack the PSP ($92.30\% + 0.57\% = 92.87\%$) ratio remains far from 100%.

The cost per test obtained by our method is much more better than the Pfahringer's winning entry [4] by performing a $CPT = 0.2228$.

In our knowledge, there is not any work in the literature that has exceeded the Pfahringer's [4] winning entry.

Most R2L instances are predicted as normal connections. In the following, we explain why this class is misclassified in the normal type. The main reason is the transformation done over DARPA98 to obtain KDD99 where most attacks

of type R2L in the test data set are not different from many normal connections in the training data set.

In order to construct valuable behavior models, many features should be gathered to characterize the considered behavior. However, the raw "unstructured" data collected from a network or other sources are not easy to analyze by different classification techniques which need more structured data format to work well. A data preprocessing phase of the gathered raw data must be performed to extract meaningful features and measures.

5 Why KDD99 is not an appropriate transformation?

The intrusion detection database KDD99 is a result of a transformation, into connection records using some data mining techniques [6], of a *tcpdump* traffic DARPA98 collected in a local area network, during nine weeks, simulating a typical U.S. Air Force LAN. The MIT Lincoln Laboratories operated this simulated LAN as if it were a true Air Force environment, but peppered with the 39 different attacks types. However, we should mention that the transformation done in MADAM ID [6] presents some drawbacks due to some limitations of the tools used for this task and the lack of some basic definitions and necessary conditions that must be satisfied by this transformation.

In the following, we introduce some definitions and conditions that a good transformation should satisfy without losing meaningful information from the initial form of the data.

The transformation task may be formalized as the following. Let R be the raw data set collected from the network traffic or other sources depending on the environment we are interested in analyzing to discover known or new computer security attacks. We can formalize audit data preprocessing by a transformation function T from the raw data set R to a well featured data item set I. This last data set denotes the whole possible values of the different considered features. An item x of I is a vector of the form $(v_1, v_2, ..., v_n)$ where each value v_i is either discrete or continuous. Let $C = \{c_1, c_2, ..., c_m\}$ be a set of the different known classes to which a behavior (an item) may fall.

The classification function, that we denote F, is then used to assign a class label to an input item vector.

5.1 What is an appropriate transformation function

1. The transformation model which consists in transforming the raw data set into their corresponding items in I should be *rich* enough to distinguish between the different behaviors in the new feature space after transformation. A *poor* transformation model T may occur when some attribute values are the same in different data items that have different class labels. This means that if we consider $r_i, r_j \in R$ and $T(r_i) = x_i, T(r_j) = x_j$ where $x_i, x_j \in I$ then if $F(r_i) \neq F(r_j)$ the transformations of r_i and r_j should be different, i.e $x_i \neq x_j$. If this is not the case then the data items that share the same

attribute values but have different class labels are considered as noise data and their number must be reduced so that accurate classification models may be learned from I.

2. If two items $T(r_i) = x_i$ *and* $T(r_j) = x_j$, issued from a transformation T, have two distinct classes and have similar values for all considered attributes then two cases are present:
   (a) either the set of the considered attributes issued from transformation T is not sufficient to characterize and then differentiate them; i.e the transformation function T is *poor*. Then we should add new attributes that can render this transformation *rich*, hence the problem is resolved. More formally: if $r_i \neq r_j$, $real\_class(r_i) \neq real\_class(r_j)$ and $T(r_i) = T(r_j)$; then the function T is poor. If this case occurs then the corresponding records present incoherence with the real traffic. Therefore, the number of attributes which is not sufficient should be increased to distinguish the two distinct records in the new feature space.
   (b) or we cannot distinguish between the raw traffic of the two connections r_i et r_j having two distinct classes. In this case we cannot find a transformation function T that may distinguish the two connections form in the new feature space. More formally: If $r_i = r_j$ and $real\_class(r_i) \neq real\_class(r_j)$ then $\nexists\ T$ such that $T(r_i) \neq T(r_j)$.
   This last case is possible if we consider a subject b that knows the password of another subject a. The generated data by the intruder b who is using the account of the user a, during a *telnet* authentication session for example, would not be different from that data generated by the legitimate user a. In this situation, there is no intrusion detection method that can find this intrusion without using any additional information.

In the following, we verify the satisfaction of these different necessary conditions on the transformation performed by W. Lee et al. [6] and demonstrate that it is not the case and some attacks, having high occurrence number, belonging to the $R2L$ class do not satisfy the first condition presented above. The bad classification of class $R2L$ is particularly due to the transformation performed over DARPA98 data sets.

5.2 Discussions

In this section, we show that the different KDD99 data sets issued from the transformation T implemented in MADAM/ID [6] is *poor* and the high false negative rate for the R2L class is due to this poor transformation.

In the following, a comparative study between the confusion matrices obtained in two tests, is presented, where in the first case we use the default training data set of KDD99 as the training data set and in the second test we use the test data set as the training set. In each test, we examine the percentage of successful prediction (PSP) using the learning data set of each test as a test set. The objective of this analysis is to help us discover whether the two databases are coherent. Therefore, the different prediction ratios of the different databases

may help us to find out whether the transformation done by W. Lee et al. [6] is *poor* or not.

The learning data set coherence Let us examine now Table 3 that corresponds to the confusion matrix obtained from initial learning database when using our enhanced C4.5 induction decision trees algorithm.

Predicted Actual	%Normal	%Probing	%DoS	%U2R	%R2L	**%New**
Normal(97,278)	**99.94**	0.01	0.00	0.00	0.00	**0.05**
Probing (4,107)	0.17	**99.78**	0.00	0.00	0.00	**0.05**
DoS (391,458)	0.00	0.00	**99.99**	0.00	0.00	**0.01**
U2R (52)	1.92	1.92	0.00	**90.39**	0.00	**5.77**
R2L (1,126)	0.62	0.00	0.00	0.09	**98.93**	**0.36**
$PSP = 99.99\%$						

Table 3. Confusion matrix obtained using the enhanced C4.5 algorithm on the initial KDD99 learning database.

We notice that the different classes are predicted with high rates using the learning database to construct the tree and to generate the different rules. The successful prediction ratio is $PSP = 99.99\%$. However, the lowest prediction ratio is that of the $U2R$ classes because there are not enough instances (52) of this class in the learning set. Our enhanced C4.5 algorithm (see Table 3) has proven its ability to classify the least frequent classes, which are not covered by any of the rules generated by the decision tree algorithm, as novel attacks rather than as normal traffic.

In the field of supervised machine learning techniques, a method is said powerful if it learns and predicts easily the different instances of the training set with a low error detection and then generalizes its knowledge to predict the class of new instances. Unfortunately, the results obtained in Table 3, C4.5 induction algorithm has efficiently learned the different instances of the training set but could not classify new instances, for the moment, into their appropriate category (see for instance Table 1).

The confusion matrix presented in Table 1 shows that the two classes U2R and R2L are badly classified into the normal class. We have expected this result because the standard C4.5 is not designed to detect novel classes that are not present in the training set. We have improved this algorithm to handle these new instances but the R2L class, as showed in Table 2, remains badly classified. Hence, three cases are possible; either the enhanced algorithm failed to detect these new attacks or some KDD99 data are false or some conditions presented in Section 5.1 are possibly not satisfied. If the second case is true, then these data should be analyzed to verify their exactitude. The first assumption is not possible because if a new instance is totally distinct from all other instances of the learning set if there is not any rule issued from the decision tree generated from the learning set that can classify it (but the default rule that can classify it as a *new* instance with the enhanced algorithm) else it is not totally distinct and then belongs certainly to a known class.

The results of Table 2 showed that the enhanced C4.5 algorithm detected more new attacks of type U2R in the test data set. However, the new R2L attacks are predicted as normal connections. We have investigated those new attacks of type R2L that are predicted as normal traffic. There are exactly 7 new R2L classes namely {*named, sendmail, snmpgettattack, snmpguess, worm, xlock, xsnoop*}. Most of these attacks are predicted as normal.The false negative rate is about 99.10% (resp. 99.97%) using the enhanced C4.5 algorithm (resp. the standard C4.5 algorithm). We focus on two of them; *snmpgettattack, snmpguess* since they present 74.79%(12108/16189) out of the R2L attacks in the test data set. All instances of these two attacks are predicted as normal connections (within R2L class in Table2).

These results show that these new R2L connections are not *distinct* from the normal connections issued after transformation.

In the following paragraph, we investigate the test data set from which we construct a decision tree in order to see if it is coherent and whether the new R2L instances are classified as normal or new from the tree generated by the test data set. After the test, we may conclude that our hypothesis of transformation done over DARPA98 to KDD99 is poor and should be improved.

The test data set incoherence In this second test, we invert the two databases. Hence, the learning database consists of $311,029$ connections and the test database contains $494,021$ connections.

Using the standard and the enhanced C4.5 algorithms, we obtained the confusion matrix presented in Table 4 of the learning instances classification for this second test.

Predicted Actual	%Normal	%Probing	%DoS	%U2R	%R2L	**%New**
Normal(60,593)	**98.34**	0.02	0.03	0.01	1.50	**0.11**
Probing (4,166)	0.19	**99.35**	0.07	0.00	0.00	**0.38**
DoS (229,853)	0.01	0.00	**99.99**	0.00	0.00	**0.00**
U2R (228)	2.19	0,00	0.00	**96.93**	0.00	**0.88**
R2L (16,189)	36.40	0,02	0.01	0.05	**63.33**	**0.19**
$PSP = 97.70\%$						

Table 4. Confusion matrix relative to five classes using the rules generated by the enhanced C4.5 algorithm over the learning database of the second test.

Although the percentage of successful prediction rate, from confusion matrix 4, is $PSP = 97.70\%$, it is considered very low since it consists in classifying the labeled (known) instances of the learning data set. This means that the C4.5 algorithm failed to learn instances with their appropriate labels. This rate is considered very low in the machine learning domain because it could not learn the instances whose classes are known a priori. On the other hand, The R2L class is highly misclassified. The classifier has learned only 63.33% from all the R2L labeled instances.

Most misclassified R2L instances are predicted as normal connections. This result justifies our observation stated in the above subsection i.e the new R2L attacks are not distinct from the normal connections, issued after transformation.

We investigated the different ratios of misclassified attacks of type R2L, we find out that these misclassified attacks are of type *snmpgettattack* or *snmpguess*.

The *snmpgetattack* type is the most frequent class type present in the *R2L* category $(7,741/16,189)$. The decision rules generated from the decision tree constructed from the second database could not classify 71.85% of *snmpgetattack* instances that correspond to $5,562$ instances; this presents a high false negative rate. Then the test data set of KDD99 is considered incoherent.

In this case, it is not interesting to test the new test database of the second test since the learning set is not learned.

From this, we are sure that these data are false or poor due to the transformation function done by W. Lee et al. in the MADAM/ID tool [6].

The new two attacks *snmpguess* and *snmpgetattack* that are present only during the two test weeks correspond in reality to an attack scenario. In this scenario an attacker guesses the SNMP community password and then remotely monitors the router activity. The SNMP password is set to "public" by default, and is often never changed from this default value.

In the DARPA98 and then KDD99, the SNMP community password remains by default (*"public"*). Hence, during the first day of the first test week, there is an attack, against an internal router of the SNMP community password by sending SNMP requests to that router using different passwords until receiving a response from that router indicating that the password is correct. This attack is similar to the dictionary attack for password guessing. We should mention that there were more than $30,000$ SNMP requests, in the DARPA98 *tcpdump* traffic, to find out the correct password. This attack corresponds to *snmpguess* that is considered as an *R2L* attack presenting 26.75% $(4,367/16,189)$ connections in the *R2L* class. Once the attacker has guessed the password, he may easily monitor the router without being detected. Moreover, this attacker has come back many times to monitor this community during the two test weeks by using the guessed password. The attacker monitoring traffic corresponds to the the *R2L snmpgetattack* in the KDD99 database that presents 47.82% $(7,741/16,189)$ of the whole *R2L* connections in this test database.

All instances of this last attack (*snmpgetattack*) are predicted as normal (within R2L class in Table 4). This result is expected and corresponds exactly to the situation presented in the point *2.b* in Section 5.1. Indeed, this traffic will be recognized as normal because the password is guessed by an attacker. However, the *snmpguess* category should be recognized as a new attack or as a dictionary attack like guess_passwd category. Unfortunately, there is not any attribute among the 41 attributes to test the SNMP community password in the SNMP request as it is the case with some attributes that verify if it is a root password or a guest password but this is considered only in the case of *telnet*, *rlogin*, etc. services. Hence one interesting information is lost after transformation with which we cannot distinguish the traffic generated by the *snmpguess* attack with

the normal traffic. This situation corresponds exactly to the necessary condition a transformation function T should satisfy as described in point *2.a* in Section 5.1. Therefore, this transformation function is poor and some attributes should be added to differentiate some attacks using the dictionary from other traffic.

The R2L could never be predicted with high rates exceeding the Pfahringer [4] results since the transformation function T introduced by W. Lee et al. [6] is poor where normal SNMP connections are similar to *snmpguess* and *snmpgettattack* connections after transformation into 41 attributes.

6 Other experiments of *new* attacks detection

In this section, we verify the efficiency of the enhanced C4.5 algorithm in detecting new attacks that are not present when DARPA98 was built.

The transformation and the different programs (MADAM/ID [6]) done at the Columbia University are not available[3]. We did write our own programs that permit to transform the network traffic into connection records but respecting the different rules and conditions that should be taken into account as explained in Section 5.1. The new attacks we investigate are those flooding attacks generated by the different known DDoS tools such as Trinoo, TFN, TFN2K, etc., used during the year 2000 against many servers over Internet. The second attack category is the Slammer DoS worm that infected thousand vulnerable servers over internet in 2003.

We tried to detect the new DDoS and Slammer attacks that were not known when the DARPA98 was constructed as new attacks. Fortunately, they have been classified as DoS attacks. In reality, the traffic form generated from the DDoS agents is not different from that of DoS traffic which is already present in the DARPA98 database. We mention that there is not any signature based IDS that could detect the flooding DDoS traffic. On the other hand, a signature based IDS cannot detect the traffic generated by the Slammer worm without adding appropriate signatures in their database.

We have improved our method as the following. If a new connection is detected as new or as a known attack, we add its corresponding connection record in the learning database if there is not any connection that is similar to the current detected attack in the learning set and then remake the learning step with the presence of these new attacks in the learning database. This idea permits the C4.5 classifier learn the new attacks in an incremental fashion. However, the new connections that are classified in the new category (see instance Table 2 particularly the new R2L attacks detected as new traffic) are considered temporarily abnormal for launching an appropriate counter-measure. However, a thorough diagnosis should be performed to find out whether it is a new traffic corresponding to a new normal activity or it is a new attack that should be added to the learning database for further investigation.

[3] These programs are licensed to a company who now is developing it commercially.

7 Conclusion

In this paper, a new anomaly intrusion detection based on decision trees is investigated and tested over the KDD99 data sets and over real network traffic in real time. We have proven its efficiency and its application has exceeded the winning entry of the KDD99 data intrusion detection contest. Since the different MADAM/ID programs are not available and present many shortcomings, we have written the different programs that transform *tcpdump* traffic into connection records following some necessary conditions we defined. The objective of our contribution is threefold. The first consists in extending the notion of anomaly intrusion detection. The second is the necessity to improve machine learning methods by adding a new class into which novel instances should be classified since they should not be classified as any of the known classes in the learning data set. The third contribution consists in introducing some necessary conditions that should be verified by a rich transformation function. This last point was not taken into account during the transformation of the DARPA98 into KDD99 data sets. As a result many attacks traffic became identical to normal traffic after transformation. Our tool is written in C/C++ for GNU/GPL and works in real time. As future work, we are investigating its use with many correlation tools such as CRIM or CARDS and any other explicit or semi explicit correlation tool. Since these tools do not deal with unknown attacks, we are currently investigating their extension to handle these new attacks generated by our new anomaly detection to integrate them in the ongoing correlation attack scenarios.

Acknowledgments

This work was funded by the French ministry of research under the ACI DADDi project.

References

1. J. P. Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James. P. Anderson Co., Fort Washington, Pennsylvania, 1980.
2. Y. Bouzida and S. Gombault. Eigenconnections to Intrusion Detection. In 19^{th} *IFIP International Information Security Conference (SEC'2004)*, pages 241–258, Toulouse, France, August 2004. Kluwer Academic Publishers.
3. DARPA Intrusion Detection Evaluation. Available at: http://www.ll.mit.edu/IST/ideval/data/data_index.html, 1998.
4. C. Elkan. Results of the KDD'99 Classifier Learning. *ACM SIGKDD*, 1:63–64, 2000.
5. S. Hettich and S. D. Bay. The UCI KDD Archive. Available at: http://kdd.ics.uci.edu/, 1999.
6. W. Lee and S. Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. *ACM Transactions on Information and System Security*, 3(4), November 2000.
7. J. R. Quinlan. *C4.5: Programs for machine learning.* Morgan Kaufmann Publishers, 1993.