Network Intrusion Detection by Combining One-Class Classifiers

Giorgio Giacinto, Roberto Perdisci, and Fabio Roli

Department of Electrical and Electronic Engineering,
University of Cagliari, Piazza d'Armi - 09123 Cagliari, Italy
{giacinto, roberto.perdisci, roli}@diee.unica.it

Abstract. Intrusion Detection Systems (IDSs) play an essential role in today's network security infrastructures. Their main aim is in finding out traces of intrusion attempts alerting the network administrator as soon as possible, so that she can take suitable countermeasures. In this paper we propose a *misuse-based* Network Intrusion Detection architecture in which we combine multiple one-class classifiers. Each one-class classifier is trained in order to discriminate between a specific attack and all other traffic patterns. As attacks can be grouped in classes according to a taxonomy, for each attack class a number of one-class classifiers are trained, each one specialized to a specific attack. The proposed multiple classifier architecture combine the outputs of one class classifiers to attain an IDS based on *generalized attack signatures.* The aim is in labelling a pattern either as normal or as belonging to one of the attack classes according to the adopted taxonomy. The potentials and effectiveness of the proposed approach are analysed and discussed.

Keywords: Computer Security, Pattern Recognition.

1 Introduction

Intrusion Detection Systems (IDSs) are an important component of a defence-in-depth network security infrastructure. IDSs collect and analyse audit data looking for anomalous or intrusive activities. As soon as a suspicious event is detected an alarm is raised, so that the network administrator can react by applying suitable countermeasures. With respect to the source of the collected data, IDSs are divided into host-based and network-based. Host-based detectors collect audit data from operating systems facilities, application logs, file system information, etc., whereas network-based detectors collect data from packets crossing a network segment. IDSs can be further subdivided in two categories with respect to the implemented detection technique, namely misuse-based, and anomaly-based IDSs. If we view the intrusion detection problem as an instance of the generic signal-detection problem, we can consider the attacks as the signal to be detected and the normal activities as the noise [1]. Misuse-based (a.k.a. signature-based) detectors base their decisions on signal characterization, whereas anomaly-based detectors base their decisions on noise characterization. Accordingly, in order to detect an attack, a misuse-based IDS must possess a description of the attack which can be matched to the attack manifestations (i.e. the signal). Such a description is often called *attack signature*. Conversely, anomaly-based detectors rely on the assumption that attack manifestations are somehow distinguishable from the users' normal activities (i.e. the noise). Therefore, for an anomaly-based IDS to detect an attack, it must possess a model of the users' normal behaviour profile which can be compared to the attack manifestations. Despite research on the Intrusion Detection field has been active since 1980s, a number of questions about their effectiveness still remain unanswered [2]. This is mainly due to the difficulties in constructing both attack signatures and normal behaviour effective models. The misuse-based approach allows the detection of intrusions whose manifestations perfectly match the related attack signature, so that unknown (i.e. never seen before) intrusions can unlikely be detected. This problem could be solved by designing general signatures which should allow to detect at least most attack variants. Unfortunately, this is usually a hard designing task. Besides, implementing general signatures usually makes misuse-based IDSs prone to false alarms. On the other hand, anomaly-based approaches should allow the detection of both known and unknown attacks. However, due to the difficulties in choosing suitable models to characterize the normal users' activities, anomaly-based approaches usually produce a higher false alarm rate than misuse-based approaches.

It is straightforward that the trade-off between the ability to detect new attacks and the ability to generate a low false alarm rate is the key point to develop an effective IDS. Due to its ability to produce low false alarm rates, the misuse-based (signature-based) detection is currently the most widely used in enterprise environments, even at the price of a very limited ability to detect unknown attacks. In order to overcome the difficulties in detecting new attacks, a number of researchers have applied statistical pattern recognition approaches [3][4][5]. The main motivation in using pattern recognition approaches to develop advanced IDSs is their generalization ability, which may support the recognition of previously unseen intrusions that have no previously described patterns. As stated before, signature generalization could make the detector prone to false alarms. In this paper we propose a network-based Intrusion Detection System (NIDS) implemented by combining several one-class classifiers (i.e. classifiers trained using example of only one class) in order to achieve the detection of unknown attacks while keeping the false alarm rate as low as possible. Each one-class classifier is trained in order to discriminate between a specific attack and all other traffic patterns (the so called outliers). As attacks can be grouped in classes according to a taxonomy (e.g. the Weber-Kendall taxonomy [6][7]), for each attack class a number of one-class classifiers are trained, each one specialized to a specific attack. The proposed multiple classifier architecture combines the outputs of one class classifiers to attain an IDS based on generalized attack signatures. The aim is in labelling a pattern either as normal or as belonging to one of the attack classes according to the adopted taxonomy, keeping the detection rate as high as possible, the false positive rate as low as possible and recognizing as many new attacks as possible.

The paper is organised as follows. A formulation of the intrusion detection problem as an instance of the pattern classification problem is presented in Section 2. The one-class classifiers ensemble architecture is illustrated in section 3. Section 4 outlines the method used to fuse the outputs of the different one-class classifiers. The experimental evaluation of the proposed architecture using the KDD Cup 1999 data set, distributed as part of the UCI KDD Archive [8], is reported in Section 5. Conclusions are drawn in Section 6.

2 Problem Formulation

From the point of view of pattern recognition, the network intrusion detection problem can be formulated as follows: given the information about network connections between pairs of hosts, assign each connection to one out of N data classes representing normal traffic or different categories of intrusions. Computer attacks can be carried out using several techniques, each one trying to exploit operating system or application flaws. Several attack taxonomies have been proposed in the literature. An attack taxonomy aims at grouping the attacks according to a given criteria. For example, the Weber-Kendal taxonomy [6][7] groups the attacks in four classes, each class containing attacks which share the final goal, even if distinct attacks in a group are carried out using different techniques. The term connection refers to a sequence of data packets related to a particular service, e.g., the transfer of a web page via the http protocol. As the aim of a network intrusion detector is to detect connections related to malicious activities, each network connection can be defined as a pattern to be classified. Extraction of suitable features representing network connections is based on expert knowledge about the characteristics that distinguish attacks from normal connections. These features can be subdivided into two groups: features related to the data portion of packets (called payload) and features related to the network characteristics of the connection, extracted from the TCP/IP headers of packets [9]. The latter group of features can be further subdivided into two groups: intrinsic features, i.e., characteristics related to the current connection, and traffic features, related to a number of similar connections. Therefore, the following three feature sets can be used to classify each connection [10]:

- *content features*, i.e., features containing information about the data content of packets (payload) that could be relevant to discover an intrusion, e.g., errors reported by the operating system, root access attempts, etc.
- network related features
  - *intrinsic features*, i.e., general information related to the connection. They include the duration, type, protocol, flags, etc. of the connection;
  - traffic features, i.e., statistics related to past connections similar to the current one e.g., number of connections with the same destination host or connections related to the same service in a given time window or within a predefined number of past connections.

This feature categorisation is general enough to take into account the high number of features that can be used to describe network traffic.

3 A Multiple One-Class Classifier System

3.1 One-Class Classification

In conventional supervised classification problems we have example patterns from each of the classes of objects we would like to recognize. For example, in a two-classes classification problem the training set is made of a number of patterns from class A as well as a number of patterns from class B. Learning data labelled either as A or B are usually equally balanced. There exist problems for which it is difficult to produce example patterns from both the two classes. In such problems we need classifiers which are able to learn in absence of counter-examples [11]. Therefore, a one-class classifier is trained using examples from only the target class A. During operational use, the classifier aims at discriminating between patterns to be labelled as A and patterns not belonging to class A (the so called outliers). In the literature a number of different terms have been used for this problem, among them the most usual are one-class classification, novelty detection, outlier detection.

3.2 The Proposed Multiple One-Class Classifier Architecture

Let us assume that a training set containing connection examples from N different attacks as well as from normal activities is available. As stated in section 2, attacks can be grouped according to a taxonomy, therefore connection examples related to the attacks can be grouped according to the chosen taxonomy as well. Assuming that the taxonomy groups the attacks in M classes, the proposed architecture is made up of N one-class classifiers grouped in M groups. Each classifier is trained using all the training patterns related to an attack j from a given attack group i. Therefore, classifier c_{ij} has to discriminate between patterns related to attack (i, j) and patterns related to all other traffic patterns. For each connection pattern, N output values are produced (i.e. one output for each classifier) which are fused according to a decision function $f()$. Fusing the outputs produced by the classifiers of the i-th group allows attaining a generalized signature for the attack class i. For each pattern, the ensemble overall output will be a label indicating whether the pattern is related either to normal activities or to one among the M attack classes. Details about the implemented decision function will be explained in section 4.

4 Decision Function

For each pattern processed by the classifier ensemble, an output vector $\mathbf{o}$ is produced. Each entry o_h of the vector represents the output O_{ij} of one among the N classifiers c_{ij} (Figure 1). As an example, let us assume that attack class l is represented by k_l distinct attack types in the training set. Thus, the elements of the output vector from position v ($v = 1 + \sum_i k_i$, $i = 1, .., l - 1$) to position $v + k_l$ are produced by the set of one-class classifiers c_{l1}, c_{l2},..., c_{lk_l}, i.e. those

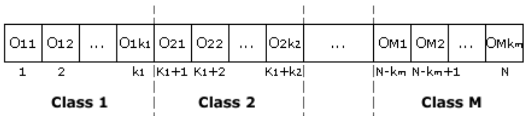

Fig. 1. Output vector

classifiers which are in charge of recognizing attacks belonging to the l-th class. The basic decision rule is straightforward. For a given test pattern, let us define

$$p = arg \max_{h=1..n} \{o_h\} \tag{1}$$

$$m = \max_{h=1..n} \{o_h\} \tag{2}$$

If $m < n_{th}$, where n_{th} is a predetermined threshold, then the pattern will be labelled as normal regardless of the position p. If $m > n_{th}$, then the position p determines the attack class the pattern belongs to. In particular, if $(v \leq p \leq v + k_l)$ then the pattern is labelled as belonging to attack class l. The threshold n_{th} is estimated by the decision template of the proposed multiple classifier system [12]. We defined the decision template as a matrix where each cell (r, s) contains the average output x of classifier c_{lj} over all the training patterns drawn from an attack class r, where the column index $s = j + \sum_i k_i$, $i = 1, .., l - 1$.

The threshold n_{th} can be tuned so that few attacks are incorrectly labeled as normal traffic. A number of additional decision thresholds can also be introduced in account of misclassification among different attack classes, and normal traffic being labeled as attacks. For example, if r refers to normal traffic and s refers to classifier c_{lj} , the value x in the cell (r, s) is interpreted as the average output of classifier c_{lj} for all the training patterns of normal traffic. If $x > n_{th}$, then the basic decision rule will probably assign a number of normal patterns to class l. To avoid this problem, we can set a new threshold $n_{lj_{th}} > x$ that will be taken into account only if m is greater than n_{th} and the position p identifies the output of the classifier c_{lj}.

5 Experimental Result

Experiments were carried out on the KDD Cup 1999 dataset distributed as part of the UCI KDD Archive [8]. The dataset is made up of a large number of network connections related to simulated normal and malicious traffic. Each connection is represented with a 41-dimensional feature vector according to the set of features illustrated in section 2. In particular, 9 features were of the intrinsic type, 13 features were of the content type, and the remaining 19 features were of the traffic type. Connections are also labelled as belonging to one out of five classes, i.e., normal traffic,Denial of Service (DoS) attacks, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Probing attacks according to the Weber-Kendal taxonomy [6][7]. Each attack class is made up of different attacks designed to attain the same effect by exploiting different vulnerabilities of the

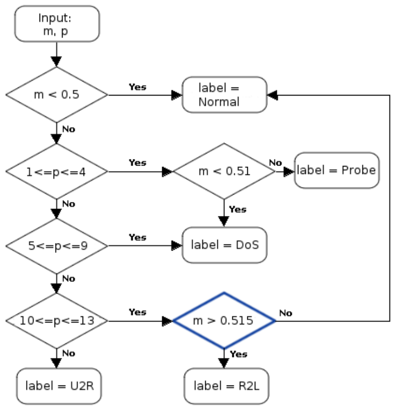

Fig. 2. Decision *schema 1* (adding "AND $N > 6$" into the highlighted rhomb we obtain the *schema 2*)

computer network. The original training set was made up of 492017 patterns related to different attacks belonging to one of the four attack categories. As the number of patterns related to the Smurf and Neptune attacks (DoS type) was very large compared to the number of patterns related to other attacks, we generated a new training set with balanced classes by pruning the patterns related to these attacks. The obtained training set contained 107701 patterns. The test set was made up of 311029 patterns, where 19091 patterns were related to attacks which belonged to one of the 4 attack classes (Probe, DoS, R2L, U2R) but which were not included in the training set. These patterns allow testing the ability of the pattern recognition approach to detect novel attack types (i.e. attacks not present in the training set). The training portion of the KDD Cup 1999 dataset contains 21 different attacks. According to the ensemble architecture described in sections 3 and 4, we have first trained 21 one-class classifiers using the k-means algorithm [11] setting k=1 and a reject percentage equal to 5%. The threshold n_{th} (section 4) has been set equal to 0.5. Each classifier has been trained using patterns related to a different attack from the training set. Afterwards, the decision template has been computed. Some of the 21 classifiers exhibited an average output grater than n_{th} for a large number of patterns related either to normal activities or to different attack classes. These classifiers were disregarded thus reducing the number of classifiers from 21 to 14 (table 1). It is worth noting that the 7 excluded classifiers were those which resulted to be heavily undertrained because of the very limited number of available training patterns. Nevertheless, the 7 attack types related to the excluded classifers may be still detected and classified as belonging to one out of the 4 attack classes thanks to the generalizion ability of the ensamble.

Performances have been first computed according to the basic decision rule described in section 4 with $n_{th} = 0.5$. Analysing the decision template we noted that some classifiers in the R2L group produced an average output greater than n_{th} when normal patterns were given as input. Besides, analysing the confusion

Table 1. Training attacks grouped according to Weber-Kendal taxonomy

Group	**Attacks**
Probe	*Ipsweep, Nmap, Portsweep, Satan*
DoS	*Land, Neptune, Ping of Death, Smurf, Teardrop*
R2L	*Ftpwrite, Guess password, Phf, Spy*
U2R	*Perl*

Table 2. Performance results

	PC%	**FP%**	**FN%**	**NA%**	**cost**
KDDCup1999 winner	92.71	0.11	6.59	7.03	0.2331
K-means - *schema 1*	86.71	6.03	1.42	80.69	0.2487
K-means - *schema 2*	88.58	1.54	6.11	20.70	0.2635

Table 3. Cost Matrix

		Assigned				
		Normal	**U2R**	**R2L**	**Probing**	**DoS**
True	**Normal**	0	2	2	1	2
	U2R	3	0	2	2	2
	R2L	4	2	0	2	2
	Probing	1	2	2	0	2
	DoS	2	2	2	1	0

matrix, we also noted that a number of patterns belonging to the DoS class were labelled as Probe. Therefore, in order to overcome this problem, we set up an additional threshold (0.515) to be used in the decision process according to the values in the decision template. The attained new decision schema is shown in Figure 2. In the following, we will refer to this schema as *schema 1*. Afterwards, a further modification to the decision criteria has been applied. We labelled a pattern as R2L only if the maximum output of classifiers trained on R2L attacks was greater then a suitable threshold (0.515), and if the number N of classifiers which produced an output higher than n_{th}=0.5 was greater than 6. In the following we will refer to this decision schema as *schema 2* (see Figure 2).

Table 2 reports the performance results in terms of the percentage of correctly classified patterns **PC**, the false positive (false alarm) rate **FP**, the false negative (missed alarms) rate **FN**, the detection rate related to new attacks (i.e. test patterns related to attack types never seen during training) **NA**, and the average classification cost computed according to the cost matrix shown in Table 3. (The cost is computed as in [13]). Results are compared to the KDD Cup 1999 winner classifier (first row in table 2). It can be easily seen that our classifier ensemble performs better than the KDD Cup 1999 winner in terms of the false negatives rate and of the new attacks detection rate, especially in case of decision schema 1. Nevertheless, the proposed ensemble performs worse than the KDD Cup 1999 winner in terms of the percentage of false positives and of the overall cost, either adopting decision schema 1 or 2. This reflects the inevitable trade-off between the false positive rate and the attack signature generalization ability.

6 Conclusions

In this paper, we have proposed a classifier ensemble architecture composed by one-class classifiers specialized in discriminating between patterns related to a specific attack class and patterns related to something else (i.e. patterns related either to normal usage or to different attack types). Combining the output of those one-class classifiers specialized in recognizing attacks belonging to a given attack class allowed us to attain an IDS based on *generalized attack signatures.* The proposed approach aimed at constructing a misuse-based Network Intrusion Detection System (NIDS) having a higher generalization ability than conventional signature-based NIDS. Experiments were carried out on the KDD Cup 1999 dataset. Performance results reflect the inevitable trade-off between the false positive rate and the *attack signature* generalization ability.

References

1. Axelsson S. *A preliminary attempt to apply detection and estimation theory to intrusion detection.* Technical report, Dept. of Computer Engineering, Chalmers Univerity of Technology, Sweden, March 2000.
2. McHugh J. *Intrusion and Intrusion Detection.* International Journal of Information Security, 2001, Vol. 1(1), 14-35.
3. Giacinto G., Roli F., Didaci L. *Fusion of multiple classifiers for intrusion detection in computer networks.* Pattern Recognition Letters, 2003, Vol. 24(12), 1795-1803.
4. Ryan J., Lin M.J., Miikkulainen R. *Intrusion Detection with Neural Networks.* In: Advances in Neural Information Processing Systems 10, 1998, M. Jordan et al., Eds., Cambridge, MA: MIT Press, 943-949.
5. Cordella, Limongiello, Sansone. *Network Intrusion Detection by a Multi-Stage Classification System.* Proceedings of the 5th International Workshop MCS 2004, 324-333.
6. D. Weber. *A Taxonomy of Computer Intrusions.* Master's thesis Massachussets Institute of Technology, 1998.
7. K. Kendall. *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems.* Master s thesis, Massachussets Institute of Technology, 1999.
8. *KDD Cup 1999 dataset.* http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
9. Northcutt S., Novak J., *Network Intrusion Detection (2 ed).* New Riders Pub, 2001.
10. Lee W., Stolfo S.J., *A framework for constructing features and models for intrusion detection systems.* ACM Trans. on Inform. and System Security, 2000, 3(4), 227-261.
11. D. Tax. *One-class classification.* PhD thesis, Technische Universiteit Delft, 2001.
12. L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley, 2004.
13. C. Elkan. *Results of the KDD 99 Classifier Learning.* ACM SIGKDD Explorations, 2000, Vol. 1(2), 63-64.