Nature-inspired Ensembles to Detect SNMP Anomalous Situations

Javier Sedano, Álvaro Herrero, Silvia González, Emilio Corchado, and Bruno Baruque

Abstract— **This experimental study is focused on the application of some nature inspired classifier ensembles for the detection of SNMP-related attacks. The reliability of the novel method proposed is compared with other models and validated under a real interesting case study, taking into account several different anomalous situations as port scan and MIB Information Transfer.**

Classifier, Ensemble, Simple Network Management Protocol, Intrusion Detection

I. Introduction

Intrusion Detection Systems (IDSs) [1-3] became an essential asset in the computer security infrastructure of most organizations. An IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack. This study deals with IDSs that monitor computer networks, that is network-based IDS.

ID has been approached from several different points of view up to now; many different nature-inspired techniques - such as Genetic Programming [4] or Neural Networks [5-7] - and some other artificial intelligent models - Data Mining [8-10], Expert Systems [11] or Fuzzy Logic [12] - have been applied to ID mainly to perform a 2-class classification (normal/anomalous or intrusive/non-intrusive). Up to now, these models have been applied to general datasets such as the DARPA [13] or KDD [14], whose limitations and problems have been previously analysed [15, 16]. To overcome such drawbacks, several datasets have been generated in a real-life network, containing real attack scenarios.

Among all the implemented network protocols, there are several of them that can be considered quite more dangerous (in terms of the network security), such as the Simple Network Management Protocol (SNMP) [17]. This protocol is a standard protocol to manage a wide range of devices on an IP network. Thus, critical information to configure such devices is managed by this protocol. SNMP was identified as one of the top five most vulnerable services by CISCO [18], specially the two first versions of this protocol that still are widely used at present time. An attack based on this protocol may severely compromise the security of the whole network [19]. SNMP attacks were also listed by the SANS (SysAdmin, Audit, Network, Security) Institute as one of the top 10 most critical internet security threats [20] [21].

Most security tools are specialized in attacks coming from the internet but attacks are just as likely to come from inside the network as from the outside. In the case of SNMP, all packets coming from outside the network would be blocked by some other security tools. However, internal attacks related with this protocol may also take place. Hence, the experimental setting of this study is focused on the identification of anomalous situations concerning SNMP.

For that purpose, biologically-inspired classifiers and some other models have been applied in this work to successfully detect some anomalous situations (that may be attacks) related to SNMP. Advancing previous work, a comprehensive collection of classifiers and ensembles has been studied to obtain the best performance. Those situations are comprehensively described in the following sections.

The remaining sections of this study are structured as follows: section 2 briefly introduces the analyzed protocol; SNMP. The applied classifiers and ensembles are described in section 3, while experimental results are presented in section 4. The conclusions of this study, as well as future work, are discussed in section 5.

II. SNMP

SNMP was oriented to manage nodes in the Internet community [17]; it is used to control routers, bridges, and some other network elements, reading and writing a wide variety of information (such as operating system, version, routing tables, default TTL and so on) about these devices. All this information is stored in the Management Information Base (MIB), so it can be defined in broad terms as the database used by SNMP to store information about the elements that it controls.

This work focus on the identification of SNMP-related attacks. The tree main anomalous situations related with this protocol are scans, SNMP community searches and MIB information transfers. These situations (described in this section) can be very risky on their own and all together (a network scan followed by an SNMP community search and ending with an MIB information transfer) make an SNMP attack from scratch. That is, an intruder gets some of the SNMP managed information without having any previous knowledge about the network being attacked.

In addition to purely SNMP anomalous situations (SNMP community searches and MIB transfers), network/port scans are also addressed in this work as it is an initial step when

Manuscript received August 25, 2011.

S. Gonzalez and J. Sedano are with Instituto Tecnológico de Castilla y León. C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001 Burgos, Spain (e-mail:j avier.sedano@itcl.es)

A. Herrero and B. Baruque are with Department of Civil Engineering, University of Burgos, Spain. C/ Francisco de Vitoria s/n, 09006 Burgos, Spain (e-mail:{ahcosio, bbaruque}@ubu.es)

E.Corchado is with Departamento de Informática y Automática, Universidad de Salamanca Plaza de la Merced, s/n, 37008 Salamanca, Spain (e-mail: escorchado@usal.es)

attacking a previously unknown network.

1) Scans

A port scan may be defined as series of messages sent to different port numbers of a host to gain information on its activity status. These messages could be sent by an intruder to find out more about the network services a host is providing. On the contrary, in a network scan the same port is the target for a number of hosts (usually all the hosts in an IP address range). A scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity. A network scan is one of the most common used techniques to identify services that might be accessed without permission [22].

2) SNMP Community Search

The unencrypted "community string" can be seen as the SNMP password for versions 1 and 2. An SNMP community search is characterized by the intruder sending SNMP queries to the same port number of different hosts trying to guess the SNMP community string by means of different strategies (brute force, dictionary, etc.) [21]. Once the community string has been obtained, all the information stored in the MIB is available for the intruder.

3) MIB Information Transfer

This situation is a transfer of some (or all the) information contained in the SNMP MIB, generally through the get (or get-bulk) command. This kind of transfer is potentially a dangerous situation. However, the "normal" behaviour of a network may include queries to the MIB. It is the network administrator responsibility to decide whether it is a "normal" MIB transfer (scheduled in advanced) or it is not.

III. Classifiers and Ensembles

As previously explained, one of the most interesting features of IDSs is their capability to automatically detect whether a portion of the traffic circulating the network is an attack or, on the contrary, it is normal traffic.

Automated learning techniques are algorithms specifically designed for the purpose of deciding about new presented data. Usually, that kind of algorithms suffer from common problems, such as the over-fitting to the data used for training - and therefore, consequent poor generalization capabilities -, the stuck on local minima in their learning function or a high computational complexity when dealing with complex data. One of the most widespread and useful techniques in order to avoid such problems is the ensemble learning scheme [23-26]. The main idea behind this kind of meta-algorithms is to train several slightly different simpler classifiers and combine their results in order to improve the results obtained by a single, usually more complex, one [27].

In the present study several of these algorithms have been considered both for the base classifiers and for the ensemble training in order to have a significant wide array of possible algorithms to compare their performance results on mutated data sets.

Among the base classifiers, it should be mentioned clustering algorithms such as the k-Nearest Neighbours (IBK) [28], instance-based statistical classification algorithms such as the Simple Classification and Regression Decision Tree (CART) [29] and the REP-Tree [30] and artificial neural-network such as the Radial Basis Function Network [31].

Among the ensemble meta-algorithms that make use of the previous mentioned simple algorithms, the test performed have made use of basic algorithms such as the Multi-Class Classifier [32], used to adapt binary classifiers to multi-class problems, Bagging [33], Adaptive Boosting (AdaBoost) [34], or Random Forest [35]. This kind of ensemble algorithms is based on the training of their base classifiers using different random selections of the initial dataset. This approach aims to increase the diversity in the ensemble, by specializing classifiers in overlapping regions of the data space. In the case of the Bagging, random samples of the dataset are selected to individually train each of the composing classifiers. The Adaboost algorithm uses a more thorough approach, as it gives a certain weight to each sample; according to how well it is recognized by the already trained classifiers. That way, the currently trained classifier can concentrate in difficult samples for previously trained ones; in an additive way. Random Forest selects both the samples and the dimensions or characteristics of the data in the training of its base classifiers (decision trees).

The results of these simple meta-algorithms are compared with more modern boosting algorithms such as the LogitBoost [36] or the StackingC [37]. The former uses a similar approach to the one of the Adaboost, but it relies on the binomial log-likelihood as a loss function, instead of the exponential function which underlies in the AdaBoost. The Stacking meta-algorithm uses a two-level method. In the first stage it trains different base classifiers and obtains the individual results of the classifications on the test dataset. Then, it uses that information in the second stage to train a meta-classifier that selects a different subset of the base classifiers depending on the data sample that is asked to classify.

As results prove, ensemble learning adds an important value to the data analysis task, as almost all variants consistently improve results obtained by the single classifier.

IV. Experimental Results

This section describes the experimental setting used for evaluating the proposed ensemble methods when facing SNMP-related anomalous situations. Further information on how the analyzed datasets were generated is also provided. Then, the obtained results are also detailed.

A. Datasets

Real-life datasets have been previously applied to perform ID [38, 39]. Packets travelling along the network are characterized by using a set of features extracted from the packet headers. These five features, once codified,

contribute to build up the input vectors of the machine-learning models, $\mathbf{x} \in \Re^5$. The studied features can be listed as follows:

- Timestamp: the time when the packet was sent.
- Source port: the port number of the device that sent the packet.
- Destination port: the port number of the target host, i.e. the host to which the packet is sent.
- Protocol ID: an integer number that identifies the protocol over TCP of the packet.
- Size: the packet size (in Bytes).

It has been proved that this low-dimensional datasets allow for the detection of some anomalous situations, mainly those related to SNMP [6].

Packets travelling along a medium-size university network were captured, analyzed and processed to gather the data. This dataset generation methodology has been previously described in detail [39]. Apart from the anomalous traffic, normal traffic from many other protocols running on the network has been also captured.

For a complete study, two different datasets were generated, containing different examples of SNMP anomalous situations:

- Dataset 1: it contains examples of al the three anomalous situations described in section II. All packets have been labelled according to the following classes:
    1. Normal traffic.
    2. Scans to port number 161 (SNMP default port number).
    3. Scans to port number 162 (SNMP default port number).
    4. Scans to port number 3750.
    5. MIB information transfer.
    6. Community search.
- Dataset 2: it contains examples of only two of the anomalous situations: scans and MIB information transfer.

B. Experiments

Two different datasets were used for both training and testing. The number of samples used for training and validation are 9821 and 5866, respectively. A 10-fold cross-validation schema was selected. The final classification rate is obtained using the two previously described datasets, To check the precision of the classifier ensembles, three and six classes were used. The experimental setup comprises 25 ensembles such as "Adaboost", "Random Subspaces", "Decorate", "Rotation Forest", "Bagging", "Boosting", etc., and 30 classifiers some of them are "NaiveBayes", "Ibk", "LinearRegression", "JRip", "RBFNetwork", "SMO", etc.

Each ensemble uses a combination of 10 base classifiers of the same type. The experimentation realized has a total of 780 test, that comes from the combination of ensembles -25- and classifiers -30-, adding the classifiers tests without ensembles. The best results obtained are presented in the following section.

C. Results

From Table I it can be concluded that most of the classification ensembles are able to carry out the classification of the three-classes dataset and get the right classification of 100% of the classes. This means that all the applied classifiers are able to distinguish normal from anomalous (rather scan or MIB transfer) traffic.

TABLE I
RESULTS WITH THREE CLASSES

Ensemble	Classifier	Classification Rate
FilteredClassifier	Ibk, SimpleCart, Ib1 and AODE	Training (99.98%) Classification (100%)
Adaboost	JRip	Training (99.93%) Classification (100%)
Adaboost	NaiveBayes	Training (99.95%) Classification (100%)
MultiboostAB	JRip	Training (99.93%) Classification (100%)
MultiboostAB	SimpleCart	Training (99.98%) Classification (100%)
RandomSubSpace	Ibk	Training (99.98%) Classification (100%)
RandomSubSpace	JRip	Training(99.95%) Classification (100%)
RotationForest	JRip	Training (99,95%) Classification (100%)
RotationForest	SimpleCart	Training (99.98%) Classification (100%)
Bagging	JRip	Training (99.94%) Classification (100%)

In some occasions, being able to distinguish normal from anomalous traffic is not enough, specially if automatic actions are run to immediately stopped ongoing attacks. Thus, a more precise classification of attack traffic is requested. To do so, experiments were conducted on six-classes datasets, for the IDS to be able to clearly identify each one of the anomalous situations in the dataset. The best results from this second round of experiments are presented in Table II.

TABLE II
RESULTS WITH SIX CLASSES

Ensemble	Classifier	Classification Rate
Classification Via Regression	Linear Regression and Pace Regression	Training (94.654%) Classification (88.97%)
Classification Via Regression	Simple Linear Regression	Training (92.09%) Classification (85.53%)
LogitBoost	Pace Regression	Training (95.53%) Classification (87.95%)

As can be seen in Table II, a more specific classification of attacks is not properly performed by any of the applied models and ensembles. The best classification rate of these experiments is 88.97%.

Table 3 shows the characteristics and options of the chosen ensembles, together with their tuned values.

TABLE III
SELECTED OPTIONS OF THE ENSEMBLES

Ensembles	Options
FilteredClassifier	Name of the filter "Discretize"
Adaboost	Number of boost iterations (10), seed for resampling (1), use resampling instead of reweighting (false), percentage of weight mass (100).
MultiboostAB	Number of boost iterations (10), number of sub-committees.(3), seed for resampling (1), use resampling instead of reweighting (false), percentage of weight mass (100).
RandomSubSpace	Number of iterations (10), Size of each subSpace (0.5), seed for resampling (1).
RotationForest	Maximum size of a group (3), Minimum size of a group (3), number of iterations to be performed (10), number of groups (false), filter used "Principal Components", percentage of instances to be removed (50), seed for resampling (1).
Classification Via Regression	Classifier (M5P)
Bagging	Size of each bag (100), compute out of bag error (False), number of bagging iterations (10), seed for resampling (1).
LogitBoost	Classifier (DecisionStump), threshold (-1.79), Number of folds for internal cross-validation (0), number of iterations (10), number of runs for internal cross-validation (1), seed (1), shrinkage (1.0), resampling is used (false), weight pruning (100).

V. Conclusions and Future Work

This study has proposed the combination of a great amount of classifier and ensemble methods for the detection of SNMP-related anomalous situations.

Experimental results show that some of the applied ensembles attain pretty good performance when splitting the analysed data in three different classes. On the other hand, when a more precise classification is needed, the performance is not that good, increasing the number of misclassified packets.

Future work will focus in the improvement of the ensemble strategy to improve the classification accuracy when dealing with six different classes. On the other hand, some other attack contexts, apart from SNMP will be also studied.

Acknowledgment

This research has been partially supported through the projects of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 and 020000-2009-12 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT Project also funded by the MICINN, the Spanish Ministry of Science and Innovation PID 560300-2009-11 and the Junta de Castilla y Len CCTT/10/BU/0002.

References

[1] "Computer security threat monitoring and surveillance," James P. Anderson Co 1980.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering,* vol. 13, pp. 222-232, 1987.

[3] T. Chih-Fong, H. Yu-Feng, L. Chia-Ying, and L. Wei-Yang, "Intrusion detection by machine learning: A review," *Expert Systems with Applications,* vol. 36, pp. 11994-12000, 2009.

[4] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs," *International Journal of Network Security,* vol. 4, pp. 328-339, 2007.

[5] S. T. Sarasamma, Q. M. A. Zhu, and J. Huff, "Hierarchical kohonenen net for anomaly detection in network security," *IEEE Transactions on Systems Man and Cybernetics, Part B,* vol. 35, pp. 302-312, Apr 2005.

[6] Á. Herrero, E. Corchado, P. Gastaldo, and R. Zunino, "Neural projection techniques for the visual inspection of network traffic," *Neurocomputing,* vol. 72, pp. 3649-3658, 2009.

[7] C. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks," *Pattern Recognition Letters,* vol. 26, pp. 779-791, 2005.

[8] K. Julisch, "Data mining for intrusion detection: A critical review," in *Applications of data mining in computer security*, D. Barbará and S. Jajodia, Eds.: Kluwer Academic Publishers, 2002, pp. 33-62.

[9] G. Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks," *Pattern Recognition Letters,* vol. 24, pp. 1795-1803, Aug 2003.

[10] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security,* vol. 24, pp. 295-307, 2005.

[11] H. K. Kim, K. H. Im, and S. C. Park, "Dss for computer security incident response applying cbr and collaborative response," *Expert Systems with Applications,* vol. 37, pp. 852-870, 2010.

[12] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Applied Soft Computing,* vol. 9, pp. 462-469, 2009.

[13] "Darpa intrusion detection evaluation," MIT Lincoln Laboratory, Massachusetts Institute of Technology.

[14] K. C. 1999, "Kdd cup 1999 dataset." vol. 2009, 1999.

[15] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 darpa off-line intrusion detection system evaluation as performed by lincoln laboratory," *ACM Transactions on Information and System Security,* vol. 3, pp. 262-294, 2000.

[16] S. Maheshkumar and S. Gursel, "Why machine learning algorithms fail in misuse detection on kdd intrusion detection data set," *Intelligent Data Analysis,* vol. 8, pp. 403-415, 2004.

[17] J. Case, M. S. Fedor, M. L. Schoffstall, and C. Davin, "Simple network management protocol (snmp)," 1990.

[18] "Vulnerability statistics report," Cisco Secure Consulting 2000.

[19] J. M. Myerson, "Identifying enterprise network vulnerabilities," *International Journal of Network Management,* vol. 12, pp. 135-144, 2002.

[20] S. Institute, "The top 10 most critical internet security threats - (2000-2001 archive)," 2001.

[21] S. Northcutt, M. Cooper, K. Fredericks, M. Fearnow, and J. Riley, *Intrusion signatures and analysis*: New Riders Publishing Thousand Oaks,, 2001.

[22] A. Kulsoom, C. Lee, G. Conti, and J. A. Copeland, "Visualizing network data for intrusion detection," in *Sixth Annual IEEE Information Assurance Workshop - Systems, Man and Cybernetics (SMC) , 2005*, 2005, pp. 100-108.

[23] A. J. C. Sharkey and N. E. Sharkey, "Combining diverse neural nets," *Knowledge Engineering Review,* vol. 12, pp. 231-247, 1997.

[24] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine,* vol. 6, pp. 21-45, 2006.

[25] B. Baruque and E. Corchado, "A weighted voting summarization of som ensembles," *Data Mining and Knowledge Discovery,* vol. (in press), 2010.
[26] E. Corchado and B. Baruque, "Wevos-visom: An ensemble summarization algorithm for enhanced data visualization," *Neurocomputing,* vol. "In press", 2011.
[27] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information Systems,* vol. 7, pp. 1-10, 2000.
[28] T. Bailey and A. Jain, "A note on distance-weighted k-nearest neighbor rules," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 8, pp. 311-313, 1978.
[29] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," *Wadsworth Inc., Belmont, CA,* vol. 358, 1984.
[30] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Advances in Space Research,* vol. 41, pp. 1955-1959, 2008.
[31] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural computation,* vol. 1, pp. 281-294, 1989.
[32] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research,* vol. 1, pp. 113-141, 2001.
[33] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996.
[34] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, 1996, pp. 148-156.
[35] L. Breiman, "Random forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.
[36] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics,* vol. 28, pp. 337-407, 2000.
[37] A. K. Seewald, "How to make stacking better and faster while also taking care of an unknown weakness," in *Nineteenth International Conference on Machine Learning*: Morgan Kaufmann Publishers Inc., 2002.
[38] E. Corchado, Á. Herrero, and J. M. Sáiz, "Testing cab-ids through mutations: On the identification of network scans," in *International Conference in Knowledge-Based and Intelligent Engineering & Information Systems (KES 2006)*, 2006, Springer, Heidelberg LNAI (4252) pp. 433-441.
[39] E. Corchado and Á. Herrero, "Neural visualization of network traffic data for intrusion detection," *Applied Soft Computing,* vol. 11, pp. 2042–2056, 2011.