Deliberative Agents for Intrusion Detection

Álvaro Herrero[1], Emilio Corchado[1], María A. Pellicer[1], and Ajith Abraham[2]

[1] Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio, escorchado}@ubu.es
[2] Norwegian University of Science and Technology, Norway
ajith.abraham@ieee.org

Abstract. This work describes a multiagent system incorporating some artificial intelligence techniques for intrusion detection. The proposed Intrusion Detection System (IDS) provides a network administrator with a comprehensive visualization of the network traffic. Thus, the network manager can supervise the network activity and detect anomalies at a glance. This paper describes the structure of the Mobile Visualization Connectionist Agent-Based IDS (MOVICAB-IDS). The system includes deliberative agents using a connectionist model to identify intrusions in computer networks. Some experiments dealing with anomalous situations related to the Simple Network Management Protocol are described.

Keywords: Computer Network Security, Intrusion Detection, Multiagent Systems, Artificial Neural Networks, Unsupervised Learning, Projection Methods.

1 Introduction

Intrusion Detection Systems (IDSs) are a part of the computer security infrastructure of most organizations. They are designed to detect suspect patterns by monitoring and analysing computer network events.

There have been some previous attempts to take advantage of agents and Multiagent Systems (MAS) in the field of Intrusion Detection (ID), as for example [1], [2], [3]. Different areas of Artificial Intelligence (AI), statistical and signature verification techniques have been also used to build IDSs, as for example [4], [5], [6], [7], [8]. Additionally, visualization tools have been applied for ID, some of them providing visual measurements of network traffic.

Some AI techniques have been combined (such as genetic algorithms and fuzzy logic [9], genetic algorithms and K-Nearest Neighbor (K-NN) [10] or K-NN and Artificial Neural Network (ANN) [11] among others) in order to face ID from a hybrid point of view. In some cases they provide intelligence to MAS. This work deals with the use of a dynamic multiagent architecture employing deliberative agents capable of learning and evolving with the environment. Some of the agents contained in this architecture are known as CBR-BDI agents [12] because they integrate the BDI (Believes, Desires and Intentions) model and the Case-Based Reasoning (CBR)

paradigm. These agents may incorporate different identification or projection algorithms depending on their goals. In this case, an ANN will be embedded in such agents to perform ID in computer networks.

The use of embedded ANNs in the deliberative agents of a dynamic MAS let us take advantage of some of the properties of connectionist models (such as generalization) and agents (reactivity, proactivity and sociability).

2 System Overview

The Mobile Visualization Connectionist Agent-Based IDS (MOVICAB-IDS) [8], [13], [14] has been designed to detect anomalous situations taking place in distributed computer networks. The proposed MAS incorporates different types of agents; some of them are reactive agents while others are CBR-BDI agents.

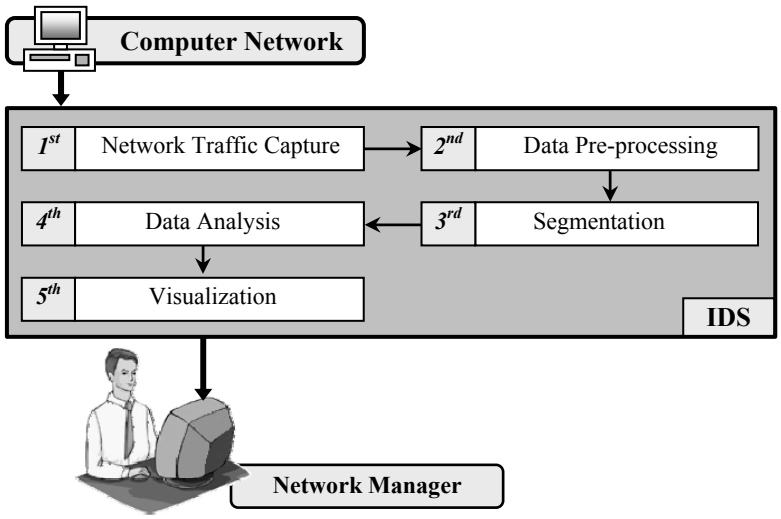


Fig. 1. Overview of the proposed IDS.

As can be seen in Fig. 1, five different tasks must be performed by the proposed IDS in order to detect anomalous situations:

- 1st task - Network Traffic Capture: all the packets travelling along the network or network segments are captured.
- 2nd task - Data Pre-processing: the captured data is selected and pre-processed. A set of packets and features contained in the headers of the captured data is selected from the raw network traffic.
- 3rd task - Segmentation: the continuous data stream is split into segments for the sake of efficiency.
- 4th task - Data Analysis: a connectionist model is applied to analyse the data.
- 5th task - Visualization: the projections are presented to the network administrator.

MOVICAB-IDS includes deliberative CBR-BDI agents [15], [16] using CBR systems [17] as a reasoning mechanism, which allows them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings. These agents use a CBR architecture, that allows them to respond to events, to take the initiative according to their goals and to make use of past experiences to find the best information to achieve goals. The CBR-BDI agents work at a high level with the concepts of Believes, Desires and Intentions (BDI) [18]. CBR-BDI agents have learning and adaptation capabilities, what facilitates their work in dynamic environments. In this work, these agents use an ANN to identify intrusions in computer networks. The following section outlines the MAS architecture and the reasoning process of the CBR-BDI agents.

3 The Multiagent System

The extended version of the Gaia methodology [19] has been applied to develop the MAS. Consequently, some roles and protocols where identified after the Architectural Design Stage. Examples of protocols are the NegotiateAnalysis protocol (when new data is ready for analysis, an analyzer is chosen) and the ChangeAnalysisConfig protocol (when the configuration of the analysis has be changed, the new configuration is sent to the CONFIGURATIONMANAGER). The Detailed Design Stage concluded that there is a one-to-one correspondence between roles and agent classes in this system, so the agent classes finally identified are: Sniffer, Preprocessor, Analyzer, ConfigurationManager, Coordinator and Visualizer. The outcomes of Gaia methodology [19], [20] are modelled by AUML [21]. Six agents have been developed to perform ID. They all are listed, and special attention will be placed to the ANALYZER (CBR-BDI) agent.

SNIFFER
This reactive agent is in charge of capturing traffic data. The continuous traffic flow is captured and split into segments in order to send it through the network for further process. Finally, the readiness of the data is communicated. One agent of this class is located in each of the network segments that the IDS has to cover (from 1 to n). Additionally, there are cloned agents (one per network segment) ready to substitute the active ones if they fail because these agents are the most critical ones. Nothing could be done if traffic data is not captured.

PREPROCESSOR
After splitting traffic data, the generated segments must be preprocessed to apply subsequent analysis. For the shake of network traffic, it would be advisable to locate one of these reactive agents in the same host where a SNIFFER is located. By doing so, the high-volume raw data will not travel along the network. Once the data has been preprocessed, an analysis for this new piece of data is requested. The sending of this data will not overload the network as its volume is much smaller than the one of split data.

ANALYZER

This is a CBR-BDI agent. It has got embedded a connectionist model within the adaptation stage of its CBR system that helps to analyze preprocessed traffic data. This connectionist model is called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [22]. It extends the Maximum Likelihood Hebbian Learning (MLHL) model [23] that is a neural implementation of Exploratory Projection Pursuit (EPP). This agent generates a solution (or achieve its goals) by retrieving a case and analyzing the new one using a CMLHL network. Each case incorporates several features, as can be seen in Table 1.

Table 1. Representation of case features. Classes: P (Problem description attribute) and S (Solution description attribute).

Class	Feature	Type	Description
P	Segment length	Integer	Total segment length (in ms).
P	Network segment	Integer	Network segment where the traffic comes from.
P	Date	Date	Date of capturing.
P	#source ports	Integer	Total number of source ports.
P	#destination ports	Integer	Total number of destination ports.
P	#protocols	Integer	Total number of protocols.
P	#packets	Integer	Total number of packets.
P	Protocol/packets	Array	An array (of variable length depending on each dataset) containing information about how many packets of each protocol there are in the dataset.
S	#Iterations	Integer	Number of iterations.
S	Learning rate	Float	Learning rate.
S	p	Float	CMLHL parameter.
S	Lateral strength	Float	CMLHL parameter.

As it is known, the CBR life cycle consists of four steps: retrieval, reuse, revision and retention [17]. The techniques and tools used by the Analyzer agent to implement these steps are described in the following paragraphs.

Retrieval stage: when a new analysis is requested, the ANALYZER agent tries to find the most similar case to the new one. Euclidean distance is used to find the most similar case in the multidimensional space defined by the features characterizing each dataset (see problem description features in Table 1).

Reuse Stage: once the most similar case has been found, its solution is reused. This solution consists of the values of the parameters used to train a connectionist model (see solution description features in Table 1). This model is CMLHL [22].

CMLHL extends the MLHL paradigm by adding lateral connections [22], which have been derived from the Rectified Gaussian Distribution [24]. The resultant net can find the independent factors of a data set but does so in a way that captures some type of global ordering in the data set. Considering a D-dimensional input vector (**x**), and an Q-dimensional output vector (**y**), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^{D} W_{ij} x_j \; ; \; i = 1,...,Q \cdot \quad (1)$$

2. Lateral activation passing:

$$y_i(t+1) = \left[y_i(t) + \tau(b - Ay)\right]^+;\ i = 1,...,Q. \quad (2)$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^{Q} W_{ij} y_i;\ j = 1,...,D. \quad (3)$$

4. Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) |\ e_j\ |^{p-1};\ i = 1,...,Q;\ j = 1,...,D. \quad (4)$$

Where: η is the learning rate, τ is the "strength" of the lateral connections, b is a bias parameter, p a parameter related to the energy function and A is a symmetric matrix used to modify the response to the data. The effect of this matrix is based on the relation between the distances separating the output neurons.

A set of trainings (for the same CMLHL model with a combination of parameter values varying in a specified range) is proposed by tacking into account the distance between the new case and the most similar one. That is, if they are quite similar, a reduced set of trainings are going to be performed. On the contrary, if the most similar case is far away from the new one, a great number of trainings are going to be generated.

Revision Stage: the CMLHL model is trained with the new dataset and the combination of parameters values generated in the reuse stage. When the new projections (the outputs of the CMLHL model for each combination) of the dataset are ready, they are shown to the human user (the network administrator typically) through the VISUALIZER agent. The user has to choose one of these projections as the best one; the one that provides the clearest snapshot of the traffic evolution.

Retention Stage: when a projection is chosen by the user, the ANALYZER agent stores a new case containing the dataset-descriptor and the solution (parameter values used to generate this projection) in the case base for future reuse (See Table 1). ANALYZER agents share their case bases.

ANALYZER agents are clearly the most resources-consuming ones. The amount of computational resources needed to analyze the continuous data coming from different network segments is extremely high. To overcome this demand, ANALYZER agents can be located in high-performance computing clusters or in most common machines.

CONFIGURATIONMANAGER

It is worth mentioning the importance of the configuration information. The processes of data capture, split, preprocess and analysis depends on the values of several parameters, as for example: packets to capture, segment length, features to extract, etc. All this information is managed by the CONFIGURATIONMANAGER reactive agent, that is in charge of providing this information to the SNIFFER, PREPROCESSOR and ANALYZER agents.

COORDINATOR

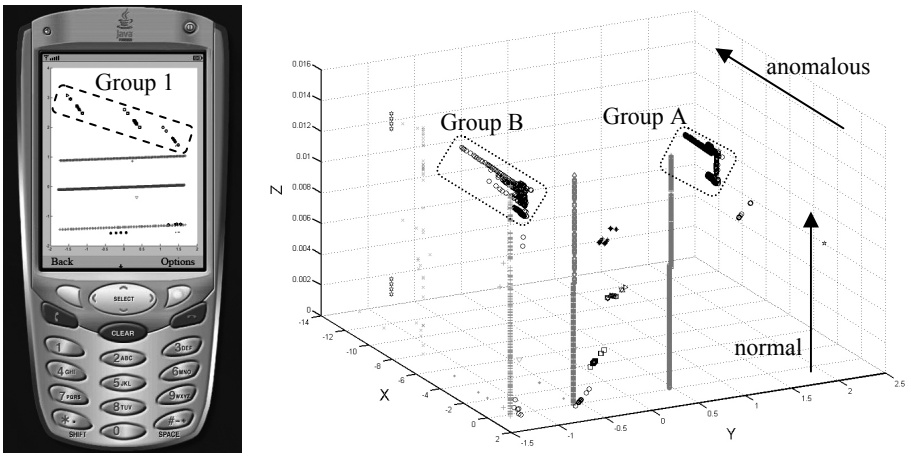
There can be several ANALYZER agents (from 1 to m) but only one COORDINATOR. The latter is in charge of sharing the analysis work out among the former. In order to

improve the efficiency and perform a real-time processing, the preprocessed data must be dynamically and optimally assigned. This assignment is performed taking into account both the capabilities of the machines where ANALYZER agents are located and the analysis demands (amount and volume of data to be analysed).

VISUALIZER
This is an interface agent. At the very end of the process, the analyzed data is presented to the person in charge of the network by means of a functional and mobile interface. To improve the accessibility of the system, the administrator may visualize the results on a mobile device (as can be seen in Fig. 2.a), enabling informed decisions to be taken anywhere and at any time. Depending on the platform where the information will be shown, the offered visualization facilities will be different.

4 Results and Conclusions

In the short-term, the Simple Network Management Protocol (SNMP) was oriented to manage nodes in the Internet community [25]. There are two main dangerous anomalous situations related to SNMP: MIB information transfers and port sweeps or scans. The MIB (Management Information Base) can be defined in broad terms as the database used by SNMP to store information about the elements that it controls. A transfer of some or all the information contained in the SNMP MIB is potentially quite a dangerous situation. A port scan may be defined as series of messages sent to different port numbers to gain information on its activity status. A port scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity.



(a) (b)

Fig. 2. Mobile (a) and advanced (b) visualizations provided by MOVICAB-IDS.

The effectiveness of MOVICAB-IDS in facing some anomalous situations has been widely demonstrated in previous works [8], [13], [14]. It identifies anomalous situations due to the fact that these situations do not tend to resemble parallel and smooth directions (normal situations) or because their high temporal concentration of packets. It can be seen in Fig. 2.a, where 3 port sweeps have been identified (Group 1) and visualized in a mobile platform. On the other hand, a more advanced visualization is offered in Fig. 2.b for a different data set. In this case, it is easy to notice some different directions (Groups A and B) to the normal data ones. Also, the density of packets is higher for these anomalous groups related to a MIB information transfer.

As a conclusion, it can be said that the MAS that incorporates CBR-BDI agents gives the following advantages:

- Scalability: new agents (both SNIFFER and ANALYZER) can be dynamically added at any time.
- Failure tolerance: backup instances of some agents can be ready to run as soon as the working instances fail, showing a proactive behaviour.
- Real-time processing: by splitting the data and allowing the system to process it in different processing units (agents located in different machines).
- Mobile visualization: the visualization task can be performed in a wide variety of devices (as it is shown in Fig. 2).

Acknowledgments. This work has been partially supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

1. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. Computer Networks: The Int. Journal of Computer and Telecommunications Networking 34(4), 547-570 (2000)
2. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. IEEE Potentials 22(4), 28-31 (2003)
3. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsettii, R.: CIDS: An agent-based intrusion detection system. Computers & Security 24(5), 387-398 (2005)
4. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised? In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 50-57. Springer, Heidelberg (2005)
5. Liao, Y.H., Vemuri, V.R.: Use of K-Nearest Neighbor Classifier for Intrusion Detection. Computers & Security 21(5), 439-448 (2002)
6. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonenen Net for Anomaly Detection in Network Security. IEEE Transactions on Systems Man and Cybernetics 35(2), 302-312 (2005)
7. Zanero, S., Savaresi, S.: Unsupervised Learning Techniques for an Intrusion Detection System. In: Proc. of the ACM Symposium on Applied Computing. pp. 412-419 (2004)
8. Corchado, E., Herrero, A., Sáiz, J.M.: Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 905-910. Springer, Heidelberg (2005)

9. Sindhu, S.S.S., Ramasubramanian, P., Kannan, A.: Intelligent Multi-agent Based Genetic Fuzzy Ensemble Network Intrusion Detection. In: Neural Information Processing. LNCS, pp. 983-988. Springer, Heidelberg (2004)
10. Middlemiss, M., Dick, G.: Feature Selection of Intrusion Detection Data Using a Hybrid Genetic Algorithm/KNN Approach. In: Design and application of hybrid intelligent systems. IOS Press. 519-527 (2003)
11. Kholfi, S., Habib, M., Aljahdali, S.: Best Hybrid Classifiers for Intrusion Detection. Journal of Computational Methods in Science and Engineering 6(2), 299 - 307 (2006)
12. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V.: Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. Expert Systems with Applications: An International Journal 34(1), 2-17 (2008)
13. Corchado, E., Herrero, A., Saiz, J.M.: Testing CAB-IDS through Mutations: on the Identification of Network Scans. In: Proc. of the Int. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems. LNAI, vol. 4252, pp. 433-441. Springer, Heidelberg (2006)
14. Herrero, A., Corchado, E., Sáiz, J.M.: MOVICAB-IDS: Visual Analysis of Network Traffic Data Streams for Intrusion Detection. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1424-1433. Springer, Heidelberg (2006)
15. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-Based Reasoning Technology. International Journal of Intelligent Systems 18(12), 1227-1241 (2003)
16. Pellicer, M.A., Corchado, J.M.: Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1), 25 - 32 (2005)
17. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39-59 (1994)
18. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A. (1987)
19. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: the Gaia Methodology. ACM Transactions on Software Engineering and Methodology 12(3), 317-370 (2003)
20. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems 3(3), 285-312 (2000)
21. Bauer, B., Müller, J.P., Odell, J.: Agent UML: A Formalism for Specifying Multiagent Software Systems. International Journal of Software Engineering and Knowledge Engineering 11(3), 1-24 (2001)
22. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. Int. Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447-1466 (2003)
23. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203-225 (2004)
24. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems 10, 350-356 (1998)
25. Case, J., Fedor, M.S., Schoffstall, M.L., Davin, C.: Simple Network Management Protocol (SNMP). RFC-1157. (1990)