Approaching Real-Time Intrusion Detection through MOVICAB-IDS

Martí Navarro[1], Álvaro Herrero[2], Emilio Corchado[3], and Vicente Julián[1]

[1] Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, Spain
{mnavarro,vinglada}@dsic.upv.es
[2] Civil Engineering Department, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
ahcosio@ubu.es
[3] Departamento de Informática y Automática, Universidad de Salamanca,
Plaza de la Merced s/n 37008, Salamanca, Spain
escorchado@usal.es

Abstract This paper presents an extension of MOVICAB-IDS, a Hybrid Intelligent Intrusion Detection System characterized by incorporating temporal control to enable real-time processing and response. The original formulation of MOVICAB-IDS combines artificial neural networks and case-based reasoning within a multiagent system to perform Intrusion Detection in dynamic computer networks. The contribution of the *anytime* algorithm, one of the most promising to adapt Artificial Intelligent techniques to real-time requirements; is comprehensively presented in this work.

Keywords Multiagent Systems, Hybrid Artificial Intelligent Systems, Computer Network Security, Intrusion Detection, Temporal Constraints, Time Bounded Deliberative Process.

1 Introduction

Softcomputing techniques and paradigms have been widely used to build Intrusion Detection Systems (IDSs) [1]. MOVICAB-IDS (MObile VIsualisation Connectionist Agent-Based IDS) has been proposed [2, 3] as a novel IDS comprising a Hybrid Artificial Intelligent System (HAIS) to monitor the network activity. It combines different AI paradigms to visualise network traffic for ID at packet level. This hybrid intelligent IDS is based on a dynamic Multiagent System (MAS) [4], which integrates an unsupervised neural projection model and the Case-Based Reasoning (CBR) paradigm [5] through the use of deliberative agents that are capable of learning and evolving with the environment. A dynamic multiagent architecture is proposed in this study that incorporates both reactive and deliberative (CBR-BDI agents [6]) types of agents. The proposed IDS applies an unsupervised neural projection model [7] to extract interesting traffic dataset projections and to display them through a mobile visualisation interface.

In other line of things, current approaches involve the application of AI techniques in real-time environments to provide real-time systems with 'intelligent' methods to solve complex problems. There are various proposals to adapt AI techniques to real-time requirements; the most promising algorithms within this field being Anytime [8] and approximate processing [9]. One line of research in Real-Time AI is related to large applications or hybrid system architectures that embody real-time concerns in many components [9], such as Guardian[10], Phoenix [11], or SA-CIRCA [12].

The MOVICAB-IDS approach can be treated as a system where its performance could be notably improved integrating real-time restrictions. Response time [13] is a critical issue for most of the security infrastructure components of an organization. The importance of a smart response on time increases in the case of IDSs. Systems that require a response before a specific deadline, as determined by the system needs, make it essential to monitor execution times. Each task must be performed by the system within a predictable timeframe, within which accurate execution of the given response must be guaranteed. This is the main reason for time-bounding the analytical tasks of MOVICAB-IDS. A key step is the assignation of each pending analysis to available ‘Analyzer agents’, which is performed by the Coordinator agent. Accordingly, temporal constraints are incorporated in the Coordinator agent that maintains its deliberative capabilities. These problems are discussed in this research in the case of the MOVICAB-IDS Coordinator Agent, which has been modelled as an agent with real-time behaviour in order to improve its performance and achieve a predictable behaviour.

This paper is organized as follows. Section 2 briefly outlines the architecture of MOVICAB-IDS. Section 3 shows how the Coordinator agent in MOVICAB-IDS is upgraded to complete an analysis before a certain deadline. To do so, the Coordinator agent integrates a temporal bounded CBR in its deliberative stage, which is comprehensively described in this section. Section 4 presents experimental results to show the benefits that arise from subjecting different phases of CBR to temporal constraints. Finally, the conclusions and future work are discussed in Section 5.

2 MOVICAB-IDS

As proposed for traffic management [14], different tasks perform traffic monitoring and ID. For the data collecting task, a 4-stage framework [15] is adapted to MOVICAB-IDS in the following way: (i) **Data capture:** as network-based ID is pursued, the continual data flow of network traffic must be managed. This data flow contains information on all the packets travelling along the network to be monitored; (ii) **Data selection**: NIDSs have to deal with the practical problem of high volumes of quite diverse data [16]. To manage high diversity of data, MOVICAB-IDS splits the traffic into different groups, taking into account the protocol (UDP, TCP, ICMP, and so on) over IP, as there are differences between the headers of these protocols. Once the captured data is classified by the protocol, it can be processed in different ways; (iii) **Segmentation**: The two first stages do not deal with the problem of continuity in network traffic data. The CMLHL model (as some other neural models) can not process data "on the fly". To overcome this shortcoming, a way of temporarily creating limited datasets from this continuous data flow is proposed by segmentation; (iv) **Data pre-processing**: Finally, the different datasets (simple and accumulated segments) must be pre-processed before presenting them to the neural model. At this stage, categorical features are converted into numerical ones. This happens with the protocol information; each packet is assigned a previously defined value according to the protocol to which it belongs.

Once the data-collecting task is performed and the data is ready, the MOVICAB-IDS process performs two further tasks: (v) Data analysis: CMLHL is applied to analyse the data. Some other unsupervised models have also been applied to perform this task for comparison purposes; (vi) Visualisation: the projections of simple and accumulated segments are presented to the network administrator for scrutiny and monitoring. One interesting feature of the proposed IDS is its mobility; this visualisation task may be performed on a different device other than the one used for the previous tasks. To improve the accessibility of the system, results may be visualised on a mobile device (such as phones or blackberries), enabling informed decisions to be taken anywhere and at any time. In summary, the MOVICAB-IDS task organisation comprises the six tasks described above.

MOVICAB-IDS has been designed, on the basis of Gaia methodology [17], [18], as a MAS that incorporates the following six agents:

- **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 processing. 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*).
- **Preprocessor**: after splitting traffic data, the generated segments are preprocessed prior to their analysis. Once the data has been preprocessed, an analysis for this new piece of data is requested.

- **Analyzer**: this is a CBR-BDI agent. It has a connectionist model embedded in the adaptation stage of its CBR system that helps to analyze the preprocessed traffic data. The connectionist model is called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [7]. This agent generates a solution (or achieves its goals) by retrieving a case and analyzing the new one using a CMLHL network.
- **ConfigurationManager**: the configuration information is important as data capture, data splitting, preprocessing and analysis depend on the values of several parameters, such as packets to capture, segment length,... This information is managed by the ConfigurationManager reactive agent, which 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 being in charge of distributing the analyses among the former. In order to improve the efficiency and perform 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 the Analyzer agents are located and the analysis demands (amount and volume of data to be analysed). As is well known, the CBR life cycle consists of four steps: retrieval, reuse, revision and retention [5].
- **Visualizer**: This is an interface agent. At the very end of the process, the analyzed data is presented to the network administrator (or the person in charge of the network) by means of a functional, mobile visualization interface. To improve the accessibility of the system, the administrator may visualize the results on a mobile device, enabling informed decisions to be taken anywhere and at any time.

3 Time-bounding the MOVICAB-IDS Coordinator Agent

CBR-BDI agents [19] integrate the BDI (Belief-Desire-Intention) software model and the Case-Based Reasoning (CBR) paradigm. They use CBR systems [5] as their reasoning mechanism, which enables them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and which gives them a large capacity for adaptation to the needs of its surroundings. 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 MOVICAB-IDS Coordinator agent, in charge of assigning the pending analyses to the available Analyzer agents, is defined as a Case-Based Planning (CBP-BDI) agent [20]. CBP [21] attempts to solve new planning problems by reusing past successful plans [22]. The Coordinator agent plans to allocate an analysis to one of the available Analyzer agents based on the following criteria:

- **Location**. Analyzer agents located in the network segment where the Visualizer or Pre-processor agents are placed would be prioritised.
- **Available resources** of the computer where each Analyzer agent is running. The computing resources and their rate of use all have to be taken into account. Thus, the work load of the computers must be measured.
- **Analysis demands**. The amount and volume of data to be analysed are key issues to be considered.
- **Analyser agents behaviour**. As previously stated, these agents behave in a "learning" or "exploitation" mode. Learning behaviour causes an Analyzer agent to spend more time over an analysis than exploitation behaviour does.

As a computer network is an unstable environment, the availability of Analyzer agents change dynamically. Network links may stop working from time to time, so the Coordinator agent must be able to re-assign the analyses previously sent to the Analyzer agents located in the network segment that may be down at any one time. As previously stated, the current version of the MOVICAB-IDS is unable to ensure the analysis of a network segment in a maximum amount of time, losing efficiency and reducing the CPU utilization capability. In order to improve the efficiency and perform real-time processing, the Coordinator agent is upgraded to become a Temporal Bounded Case-Based Planning (TB-CBP) BDI agent, bringing MOVICAB-IDS closer to real-time ID. TB-CBP is based on Temporal Bounded CBR as explained in the next section.

3.1 Temporal Bounded CBR

The Temporal Bounded CBR (TB-CBR) is a modification of the classic CBR cycle specially adapted to be applied in domains with temporal constraints. In real-time environments, the CBR stages must be temporal bounded to ensure that the solutions are produced on time; giving the system a temporal bounded deliberative case-based behaviour.

The different phases of the TB-CBR cycle are grouped in two stages according to their function within the reasoning process of an agent with real-time constraints. The fist one, called learning stage, consists of the revise and retain phases; and the second one, named the deliberative stage, includes the retrieve and reuse phases. Each phase will schedule its own execution time to support the designer in the time distribution among the TB-CBR phases. These stages can incorporate an anytime algorithm [23], where the process is iterative and each iteration is time-bounded and may improve the final response.

To ensure up-to-date cases in the case base, the TB-CBR cycle starts at the learning stage, which entails checking whether previous cases are awaiting revision and could be stored in the case base. The solutions provided by the TB-CBR are stored in a solution list at the end of the deliberative stage. This list is accessed when each new TB-CBR cycle begins. If there is sufficient time, the learning stage is implemented for cases where solution feedback has recently been received. If the list is empty, this process is omitted.

Once the learning stage finishes, the deliberative starts. The retrieval algorithm is used to search the case base and chose a case that is similar to the current case (i.e. the one that characterizes the problem to be solved). Each time a similar case is found, it is sent to the reuse phase where it is transformed into a suitable plan for the current problem by using a reuse algorithm. Therefore, at the end of each iteration in the deliberative stage, the TB-CBR method is able to provide a solution to the problem at hand, which may be improved in subsequent iterations if there is any time remaining at the deliberative stage. See more details in [24].

3.2 TB-CBR Operation within the Coordinator Agent

In the aforementioned environment, analysis planning must be completed within a maximum time. For this reason, an agent, which provide the necessary control mechanisms to carry out this task, is deployed to complete the analysis on time. Consequently, when a new segment is ready for analysis, the Coordinator agent, which is a real-time agent, has a limited amount of time to assign the pending analysis to the available Analyzer agents, which have to provide an answer as soon as possible. Therefore, a temporal constraint on the process (starting with a new generated segment and ending with the Analyzer agent giving the answer) is essential to ensure prompt execution. To perform this temporal control, all the steps in the process must be known and must be temporal bounded. Additionally, the system has to be deterministic. To guarantee these conditions, the Coordinator agent takes advantage of the TB-CBP to assign the pending analysis. So, the four phases of the TB-CBP cycle of the Coordinator agent are re-defined to comply with the temporal constraints following the TB-CBR guidelines (see [24]).

The first stage (learning stage) is executed if the agent has the plans from previous executions stored in the *solutionQueue* (these are previous executions of the CBR cycle that have not been revised and retained). The plans are stored just after the end of the deliberative stage. In this case, the following phases are executed:

- **Revise**: the plan revision consists of a two-fold analysis. On the one hand, planning failures are identified by finding under-exploited resources. As an example, the following hypothetical situation is identified as a planning failure: one of the Analyzer agents is not busy performing an analysis while the other ones have a list of pending analyses. On the other hand, execution failures are detected when communication with Analyzer agents has been interrupted. Information on these failures is stored in the case base for future consideration. When an execution failure is detected, the CBP cycle is run from the beginning, which renews the analysis request.
- **Retain**: when a plan is adopted, the Coordinator agent stores a new case containing the dataset-descriptor and the solution.

The deliberative stage is only launched if there is a new network segment to be analysed (a new pre-processed dataset is ready) by adding it to the *problemQueue* of the Coordinator agent. This will launch the execution of the following phases:

- **(Plan) Retrieve**: as previously stated, when a new pre-processed dataset is ready, an analysis is requested from the Coordinator agent. The most similar plan is obtained by associative retrieval, taking into account the case/plan description. As the time required to extract a case from the case base is predictable, the Coordinator agent knows how long it takes to get the first solution. Moreover, if the Coordinator agent has some extra time to plan the analyses, it will attempt to improve this first plan within the available time by continuing searching previously stored plans.
- **Reuse**: the retrieved plan is adapted to the new planning problem. The only restriction is that the analyses running at that time (the results of which have not yet been reported) cannot be reassigned. The others (pending) can be reassigned in order to optimize overall performance. This phase is also temporal bounded. The Coordinator agent knows when it will finish the adaptation of the cases to the new planning problem. In this phase, as the Coordinator agent calculates when the analysis agents will finish their tasks, it either knows the available time to continue building the plan. The Analyzer agents will still be executing pending analyses when this phase is completed. Thus, the assignment of an analysis to an Analyzer depends on its work load at that particular time.

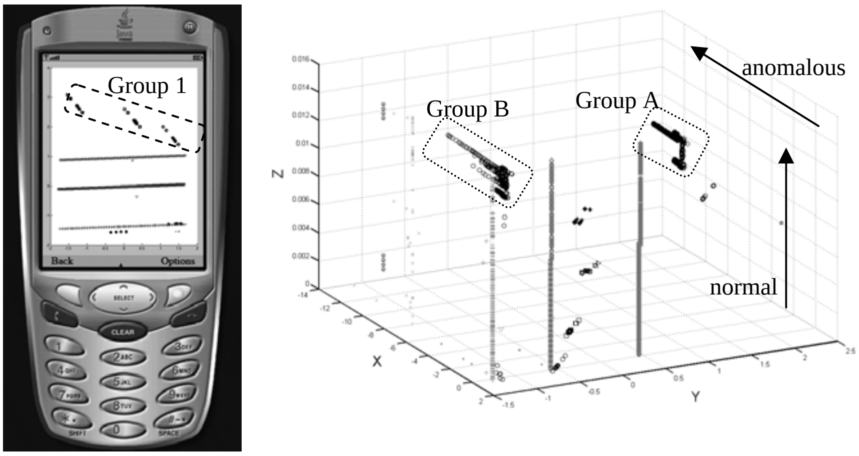
The main advantage of using the TB-CBP with regard to using a CBP without temporal constraints is to ensure a system response on time. The use of TB-CBP allows the distribution of the analysis to the Analyzer agents taking into account the available time to perform this task. On the other hand, the application of TB-CBP improves the CPU utilization and minimizes the average execution time of the analyses as it has been checked in a set of tests. The analysis requests are launched for 2 minutes following exponential distribution in which α parameter value is 0.3 (a request is generated each 3 seconds approximately). The results obtained after one hundred executions are shown in Table 1.

Table 1**.** TB-CBP vs. CBP

	CPU utilization	Analysis fulfilled on time	Average Execution Time
TB-CBP	97 %	98.2 %	1.6 ms
CBP	72 %	61.5 %	2.4 ms

4 Experimental Results: MOVICAB-IDS Visualizations

There are two main dangerous anomalous situations related to SNMP [25].: 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) (b)

Fig. 1. 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 [2, 3, 26, 27]. 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. 1.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. 1.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.

5 Conclusions and Future Work

An upgraded version of MOVICAB-IDS is presented in this paper. This version imposes temporal constraints on the deliberative agents within a CBR architecture, which enables them to respond to events in real (both hard or soft) time. In this case, the deliberative Coordinator agent, working at a high level with Belief-Desire-Intention (BDI) concepts, is temporal bounded by redefining the four phases of its CBP cycle. The consequences of temporal bounding these phases are described in this paper. As a result, the Coordinator agent will always give a solution within the available time, thereby maximizing CPU utilization and minimizing average execution time of the analyses.

Acknowledgments This research is funded through the Junta of Castilla and León (BU006A08); the Spanish Ministry of Education and Innovation (CIT-020000-2008-2 and CIT-020000-2009-12); the Spanish government (TIN2005-03395 and TIN2006-14630-C03-01), FEDER and CONSOLIDER-INGENIO (2010 CSD2007-00022). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A. for supporting the project through the MAGNO2008 - 1028.- CENIT Project funded by the Spanish Ministry of Science and Innovation**.**

References

1. Abraham, A., Jain, R., Thomas, J., Han, S.Y. (2007) D-SCIDS: Distributed Soft Computing Intrusion Detection System. Journal of Network and Computer Applications 30(1): 81-98
2. Herrero, Á., Corchado, E. (2009) Mining Network Traffic Data for Attacks through MOVICAB-IDS. (ed) Foundations of Computational Intelligence 4. Springer. Studies in Computational Intelligence 377-394
3. Corchado, E., Herrero, Á. (2010) Neural Visualization of Network Traffic Data for Intrusion Detection. Applied Soft Computing ("Accepted with changes")
4. Wooldridge, M., Jennings, N., R. (1995) Agent theories, architectures, and languages: A survey. Intelligent Agents
5. Aamodt, A., Plaza, E. (1994) Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1): 39-59
6. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V. (2008) Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. Expert Systems with Applications: An International Journal 34(1): 2-17
7. Corchado, E., Fyfe, C. (2003) Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence 17(8): 1447-1466
8. Dean, T., Boddy, M. (1988) An Analysis of Time-dependent Planning. (ed) 7th National Conference on Artificial Intelligence.
9. Garvey, A., Lesser, V. (1994) A Survey of Research in Deliberative Real-time Artificial Intelligence. Real-Time Systems 6(3): 317-347
10. Hayes-Roth, B., Washington, R., Ash, D., Collinot, A., Vina, A., Seiver, A. (1992) Guardian: A Prototype Intensive-care Monitoring Agent. Artificial Intelligence in Medicine 4: 165-185
11. Howe, A.E., Hart, D.M., Cohen, P.R. (1990) Addressing Real-time Constraints in the Design of Autonomous Agents. Real-Time Systems 2(1): 81-97
12. Musliner, D.J., Durfee, E.H., Shin, K.G. (1993) CIRCA: A Cooperative Intelligent Real-time Control Architecture. IEEE Transactions on Systems, Man, and Cybernetics 23(6): 1561 - 1574
13. Kopetz, H. (1997) Real-time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers
14. Babu, S., Subramanian, L., Widom, J. (2001) A Data Stream Management System for Network Traffic Management. (ed) Workshop on Network-Related Data Management (NRDM 2001).
15. Herrero, Á., Corchado, E. (2008) Traffic Data Preparation for a Hybrid Network IDS. (ed) Third International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2008) 5271. Springer, Heidelberg. LNAI

16. Dreger, H., Feldmann, A., Paxson, V., Sommer, R. (2004) Operational Experiences with High-Volume Network Intrusion Detection. (ed) 11th ACM Conference on Computer and Communications Security. ACM Press New York.
17. Zambonelli, F., Jennings, N.R., Wooldridge, M. (2003) Developing Multiagent Systems: the Gaia Methodology. ACM Transactions on Software Engineering and Methodology 12(3): 317-370
18. Wooldridge, M., Jennings, N.R., Kinny, D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems 3(3): 285-312
19. Pellicer, M.A., Corchado, J.M. (2005) Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1): 25 - 32
20. Bajo, J., Corchado, J., Rodríguez, S. (2007) Intelligent Guidance and Suggestions Using Case-Based Planning. (ed) Case-Based Reasoning Research and Development 4626. Springer, Heidelberg. LNAI
21. Hammond, K.J. (1989) Case-based Planning: Viewing Planning as a Memory Task. Academic Press Professional, Inc.
22. Spalzzi, L. (2001) A Survey on Case-Based Planning. Artificial Intelligence Review 16(1): 3-36
23. Dean, T., Boddy, M.S. (1988) An Analysis of Time-Dependent Planning. (ed) 7th National Conference on Artificial Intelligence.
24. Navarro, M., Heras, S., Julián, V. (2009) Guidelines to Apply CBR in Real-Time Multi-Agent Systems. Journal of Physical Agents 3(3): 39-43
25. Case, J., Fedor, M.S., Schoffstall, M.L., Davin, C. (1990) Simple Network Management Protocol (SNMP). IETF RFC 1157.
26. Corchado, E., Herrero, Á., Sáiz, J.M. (2005) Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (ed) 15th International Conference on Artificial Neural Networks (ICANN 2005) 3697. Springer, Heidelberg. LNCS
27. Corchado, E., Herrero, Á., Sáiz, J.M. (2006) Testing CAB-IDS through Mutations: on the Identification of Network Scans. In: Gabrys, B., Howlett, R.J., Jain, L.C. (ed) International Conference in Knowledge-Based and Intelligent Engineering & Information Systems (KES 2006) 4252. Springer, Heidelberg. LNAI