International Journal of Research in Computer Science
eISSN 2249-8265 Volume 4 Issue 1 (2014) pp. 11-17
www.ijorcs.org, A Unit of White Globe Publications
doi: 10.7815/ijorcs.41.2014.076

A TWO-LEVEL AUTONOMOUS INTRUSION DETECTION MODEL INSPIRED BY THE IMMUNE SYSTEM

Elnaz B. Noeparast[1], Reza Ravanmehr[2]
[1]*Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, IRAN*
Email: e-b-noeparast@iauctb.ac.ir
[2]*Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, IRAN*
Email: r.ravanmehr@iauctb.ac.ir

Abstract: *There are several methods applied to distributed system security, which have the same absolute view of the intrusion. In their view, an operation could be legitimate or intrusive, which does not have any consistency with the complicated and heterogeneous nature of distributed systems. In this paper, a two level multi-agent model is proposed whose first level determines system's unsafe behaviors based on anomaly occurrence. Then, its second level calculates the probability of system log operations effectiveness in the case of intrusion happens. If this probability is greater than the first-level prediction, the anomaly is known as intrusion, otherwise it is supposed as an unexpected legal behavior. Therefore, the false positive error probability will decrease. Also, the proposed multi-agent system utilizes the human immune system whose autonomous agents do not need maintenance and detects intrusions without relying on any other central elements, just by using their own learning and interaction capability.*

Keywords: *Distributed Systems, Intrusion Detection, Multi-Agent Systems, Immune System*

I. INTRODUCTION

In distributed systems, the intrusion detection is one of the fundamental challenges to achieve security goals because of the communication distribution and unsafe accesses. In addition to firewalls and other intrusion prevention methods and tools, we need an intrusion detection system with low false positive and negative error rate for developing a complete security in these systems [1]. In order to achieve this goal, the intrusion detection system should update information and can detect the difference between an intrusion and an unexpected legitimate behavior. Nowadays, the intrusion detection systems usually have dependence on a central element [2] along with a high false positive error rate. These systems detect intrusion based on received patterns from the central element without paying any attention toward system's behavior. Consequently, there might be a behavior or an operation which is a part of legal behavior in a special system detected as an intrusion. Also if the communication between a system and central element is disconnected or the intrusion detection could not receive its information updates, the false negative rate would increase.

Multi agent systems using autonomic computing could reach to their goals without any central element and just by interacting and exchanging information [3]. The proposed autonomic computing inspired by the operation of body parts like the immune system, and all control and security activities are done without brain activities as the central management system interface. So, it is possible to introduce a new method for dealing with the problems of intrusion detection systems which is inspired by the immune system operations.

This paper organizes as follows: in the next section, some related works of the immune system based on intrusion detection systems are reviewed. In section 3, the major process of the immune system is explained. After that, an autonomous system including agents and their components will be presented based on this process. This section provides the detailed description of our proposed model for interaction between agents and the related activity diagram. In section 5, we will show the simulation results for our proposed system and evaluate the convergence rate of network status to robust in different conditions. Finally, we finish the paper with concluded remarks and future works.

II. RELATED WORKS

There are many researches in the field of intrusion detection inspired by the immune system operation, also different model for optimization[4], clustering[5] and network security[6] are purposed based on mathematical deferential[7-10], cellular automata[11-13] and agents. Mathematical deferential and cellular automata have some problems like model complicity, computational states limitation and not being able to cell diffusion [14]. But multi-agent models have the capability of modeling state complicity and heterogenics of the real world. Since the immune system is a multi-agent system which its agents are

immune cells, this model is the best choice for modeling the immune system based intrusion detection systems. One of these models is proposed in [15] which is an artificial immune system inspired by the danger theory. In this system four types of agents (Ag agent, DC agent, TC agent and RP agent) detect intrusions through nitration with one another. Ag agent parses input information (system calls profile) to antigen format and sends them to DC agent placed in the host. When Ag agent sends a signal, DC agent analyses it and measures its danger value. If danger value of an antigen reaches to the threshold, TC agent in the central security system, measures the validation of intrusion detection. Then TC agent warns RP agent to respond to the intrusion.

The other multi-agent model is the event-based multi-agent intrusion detection model inspired by the immune system for large networks presented by Boukerche, Machado and et. al [16]. This model is based on the user signature's registrations to the operationally targeted system. Mobile agents are responsible for monitoring, distribution, storage, persistence and reactivity duty and differentiate between attacks, security violations, and several other security levels. Boukerche, Machado and et. al.[17] Also developing a real-time host-based intrusion detection model for anomaly detection using mobile agents, inspired by the human immune system. Byrski and Carvalho[18] proposed agent-based intrusion detection approach in MANETs, an artificial immune systems for anomaly detection, independent of specific routing protocols and services. Moreover, Herrero and et. al.[19] introduced an unsupervised connectionist multi agent intrusion detection system named MOVIH-IDS.

III. THE IMMUNE SYSTEM

The human body has the immune system to remove infection elements. This system contains operational elements such as lymphocytes and Antigen Presenting Cells (APC). One part of security in this system is Antigen-Antibody system based on the interactions between DC cells, B lymphocytes and T-Helper lymphocytes. DC cells detect microbes' Antigens by their receptors (TLR) and present them to B and T-Helper lymphocytes after collecting Antigens. By receiving these antigens from DC cells and co-stimulatory signals from T-Helper lymphocytes, B lymphocytes convert themselves to plasma cell in order to produce Antibodies, also some of them are converted to memory B cells to keep the memory of that microbes intrusion. [20]

IV. PURPOSED SYSTEM

Designing of a security system for distributed structures can be done through utilizing the characteristics of autonomous computing and autonomous operation of the immune system elements. Providing such a system require an autonomous multi-agent system whose agents are designed based on autonomous computing and each agent carries out some part of the immune system operation. Based on this system, each autonomous agent has four phases for self-adaption [21]. In monitoring phase, agent could be aware of internal and external environment conditions and be able to interact with other agents. The internal environment includes agents interactions and external environment consists of non-self agents which their existence is known as intrusion in artificial immune systems. Determination of self and non-self happens in Analysis phase. In Plan phase, agent considers the result of previous phases in order to plan a behavior set with maximum adaptation level with the environment condition. Flexible behavior makes an agent able to execute in the heterogeneous environments on all platforms. In Execution Phase, agent expresses the planned behavior as a reaction to the environment condition. The implicit knowledge embedded in autonomous agent architecture adds learning and experience exchanging capability to the agent characteristics.

Since system agents have learning ability, the system converges to robust status gradually. If agents have basic knowledge to handle non-self agents and system failures factors, the convergence speed will be high. Otherwise, it will be low as a result of low efficiency of the immature agents to detect intrusions and failures.

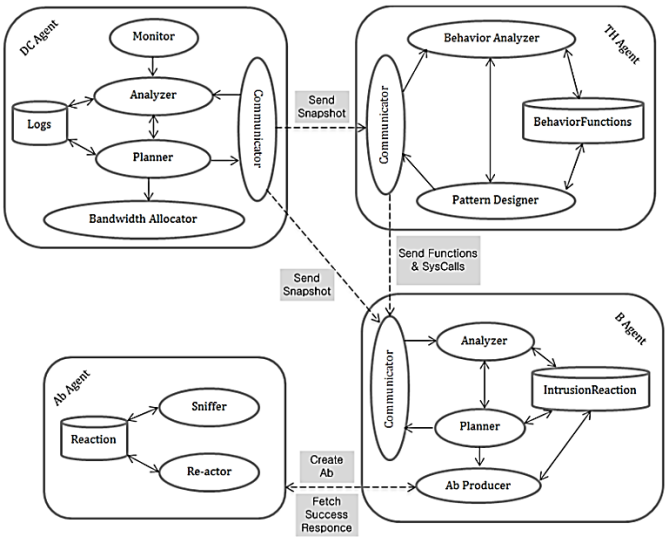

Figure 1: The proposed system (agents and their components).

The proposed model uses four kinds of autonomous agents for intrusion detection, each one has its special capabilities and is designed based on autonomic computing architecture. These agents are DC agents, TH agents, B agents and Ab agents. The structure of each agent and their communications has been presented in Figure1.

A. DC Agent

Each DC agent is placed in one of the network nodes, carries out the internal environment cognition of the node and analysis of its condition from three points of views (hardware view, software view and user view). In hardware view, the amount of main system components usage (CPU usage and memory load) and secondary components usage (network connection and bandwidth saturation) has been checked. In software view, all executing elements behavior has been studied from the point of invalid access and malicious operation. In user view the user's behavior has been analyzed.

This agent record behaviors, system statuses and operational logs of the system to be able to track them, aside with comparing current situation with the previous one through three views. Afterwards, the standard deviation value percentage is applied according to the formula as it has been shown as below. The process continues by calculating the system's failure probability percentage (δ).

$$\delta = \frac{C_H + C_S + C_U}{3} \quad (1)$$

where C_H, C_S and C_U parameter is orderly standard, deviation percent for hardware, software and user behavior. If the result is not equal to zero percent, DC agent makes a snapshot from system's condition including node ID (It can be IP or other node unique identifications), occurred (hardware, software and user) behavior information and δ. Then DC agent sends this snapshot to TH and B agent in one of neighbor nodes. This agent also allocates a dedicated bandwidth between self node in which it states and that neighbor node for arriving of Ab agent which responses to intrusions. The activity diagram of DC has been shown in Figure 2. The autonomous architecture of this agent has five modules and a knowledgebase as the following.

- *Communicator:* This module is responsible for sending snapshot to TH and B agent in one of the neighbor nodes.
- *Monitor:* This module monitors the system status from hardware, software and user views.
- *Analyzer:* This module is responsible for analyzing the system status every time and recording operational log in Logs knowledgebase.
- *Planner:* This module compares current system condition with the previous situation and calculates δ.
- *Bandwidth Allocator:* This module allocates a dedicated bandwidth between self node and neighbor node.
- *Logs:* This knowledgebase consists of operation's logs and system statuses.

B. TH Agent

TH agent carries the information regarding analyses according to recorded behavior snapshots. Then it splits them to the pattern of functions and system calls which is called signatures and sends these signatures to B agent. The activity diagram of TH agent is presented in Figure 2. This agent consists of three modules and a knowledgebase as shown as below.

- *Communicator:* This module is responsible for receiving the snapshots from DC agent and sending the signatures to B agent.
- *Behavior Analyzer:* This module tracks behaviors in the snapshots by looking through BehaviorFunction knowledgebase.
- *Pattern Designer:* Since there is different operations and system calls which might express the same behaviors, this module determines the signatures which may cause the received behavior from Behavior Analyzer.
- *BehaviorFunction:* this knowledgebase has a Column Family data model and consists of anomaly behaviors and the signatures which cause those behaviors.

C. B Agent

B agent that its operation is based on B cell in the immune system, carries out the intrusion detection and extracts suitable behaviors for responding to an intrusion. This agent receives the signatures from TH agent and the snapshot from DC agent. Then it compares these signatures with the patterns of functions and system calls in its knowledgebase. After that, considering similarity level and based on its knowledgebase, B agent calculates total intrusion probability percent (γ) in the following formula (2).

$$\gamma = \alpha \times \sum_i P_i \quad (2)$$

where α is the similarity level of received signatures with patterns in knowledgebase, i is the count of similar patterns, and P_i is the occurrence

probability percentage of a function or system call in an intrusion known as intrusion probability percentage. If γ is bigger than δ, it means that the anomaly is actually an intrusion. Hence, B agent produces Ab agent and injects the suitable behaviors and infected node ID into Ab agent's knowledgebase. The activity diagram of B agent has been shown in Figure 2. The architecture of this agent has four modules and a knowledgebase as shown as below.

- *Communicator:* This module receives the snapshots from DC agent and the signatures from TH agent.
- *Analyzer:* This module analyzes received signatures and compares them to its knowledgebase patterns, then determines the intrusion probability percentage.
- *Planner:* this module is responsible for detecting the accuracy of an intrusion and deciding on producing the suitable behaviors in order to respond. If γ is less than δ, the anomaly is assumed as an unexpected legitimate behavior and no response is produced. But if γ is bigger than δ, it is assumed that the knowledge of B agent is not up-to-date. Hence, this module learns signatures that caused the anomaly, which does not exist in the knowledge base. Then inserts δ as the intrusion probability percentage to its taught pattern knowledgebase. After that, it sends the infected node ID and the behaviors for responding to intrusion to Ab Producer module.
- *Ab Producer:* This module produces an Ab agent and injects received ID and the response behaviors in Ab agent's knowledgebase. This module also fetches the successful behavior (which removes the effect of the intrusion in the infected node) of Ab agent after its returning from the infected node and increases the rank of that behavior in the knowledgebase.
- *Intrusion Reaction knowledgebase:* This knowledgebase consist of two sets of knowledge. Firstly, the intrusion knowledge includes the pattern of functions and system calls, which may help an intrusion, also the intrusion probability percentage for each pattern. Secondly, the intrusion response behaviors based on the patterns in it, which is measured based on their usage in responding to an intrusion.

D. AB Agent

Ab agent has similar specifics characteristics to the immune system antibody and it is produced for in order to respond to a specific intrusion. This agent migrates to infected cell through dedicated bandwidth and responds to intrusion when it is produced. Afterwards, it comes back to neighbor node. This agent has two modules and a knowledgebase as the fallowing.

- *Sniffer:* This module is responsible for finding the infected neighbor node with the same ID as the ID in its knowledgebase.
- *Re-actor:* This module responds to an intrusion and removes its effect. If one of intrusions responses does not remove the effects, it tries another intrusion response based on its knowledgebase. The response removes the intrusion effect and it will be labeled as a successful response.
- *Reaction knowledgebase:* This knowledgebase includes the intrusion responses and infected node ID.

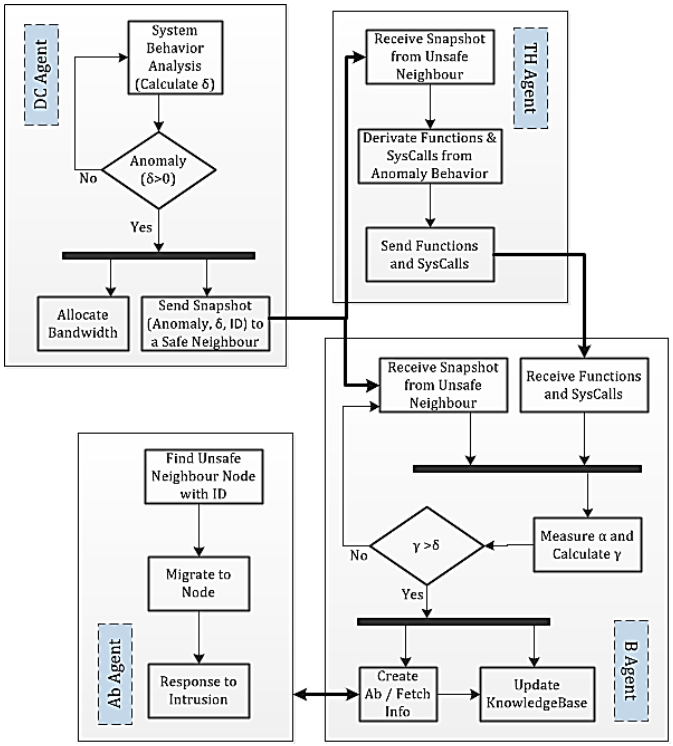

Figure 2: Agents' Activity Diagram

V. SIMULATION RESULTS

In order to simulate the proposed model, NetLogo has been used which is an agent-based simulation environment [22]. In Figure 3, a distributed system is designed and each node has a direct connection with some other nodes that are called neighbors. Each node experiences three conditions in the network life cycle which are safe, suspicious and unsafe. These conditions are shown orderly by green, yellow and red. When an anomaly condition happens to a node, a dedicated bandwidth is allocated between this node and one of the safe neighbor nodes. This action is equivalent with the proposed model, when DC agent calculates system failure probability percentage (δ) and sends snapshot to TH and B agents in one of neighbors. Then based on TH and B agent operation, δ is compared with the total intrusion probability percentage (γ). If γ is less than δ, the node's color changes from green to yellow and the node would be

known as a suspicious node in which an unexpected legitimate behavior is occurred. But if γ is bigger than δ, the node's color changes to red.

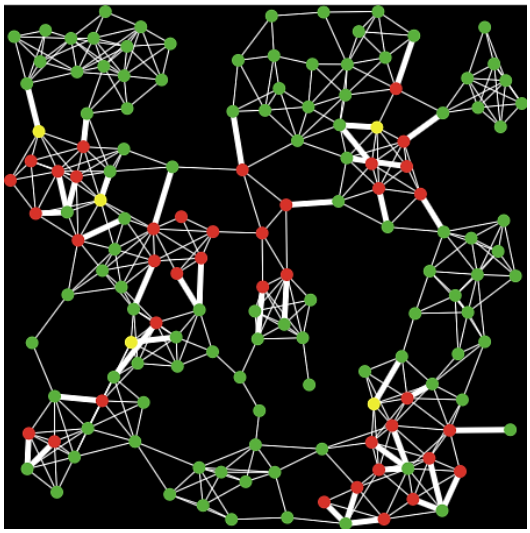

Figure 3: The Simulation Environment

In order to aiming to respond to unsafe nodes, an intrusion response probability percentage (ω) is considered. This percentage is the results of intrusion response ranks in B agent knowledgebase which shows the ability of producing Ab agent for responding to an intrusion and changing the status of node from unsafe to safe. When agents are immature, this percent is low because they do not have enough knowledge to detect and express a suitable response to intrusions. On the other hand, when these agents are mature, this percentage is high. Immature agents gradually change to mature agents using their learning ability, along with this process ω increases too. Therefore, if the probability percentage of changing from unsafe to safe status is less than ω, it means B agent in neighbor node does not have enough knowledge to present a suitable response to the intrusion. Therefore the node status does not change and its DC Agent tries another neighbor node's agents to interact with. But if this probability percentage is bigger than ω, node status changes to safe and ω increase.

Based on Figure 4 that shows the network status in several conditions (considering ω and infection spread percentage (ε)). If ω and ε are low, the number of immature agents is not very tangible, also reaching to a robust status for the network would be time consuming. When ε increases, the number of immature agents is tangible, but because of the learning ability of these agents network status would converge to a robust status in a long period of time. If ε reaches to its maximum value there is not enough time for immature agent to learn, so they cannot converge to a robust network status. In this situation, the suspicious and unsafe nodes condition will not change permanently. When half of agents are mature, they cause that the status of network is converged to robust, although it takes a longer time than a network with more matured agent.

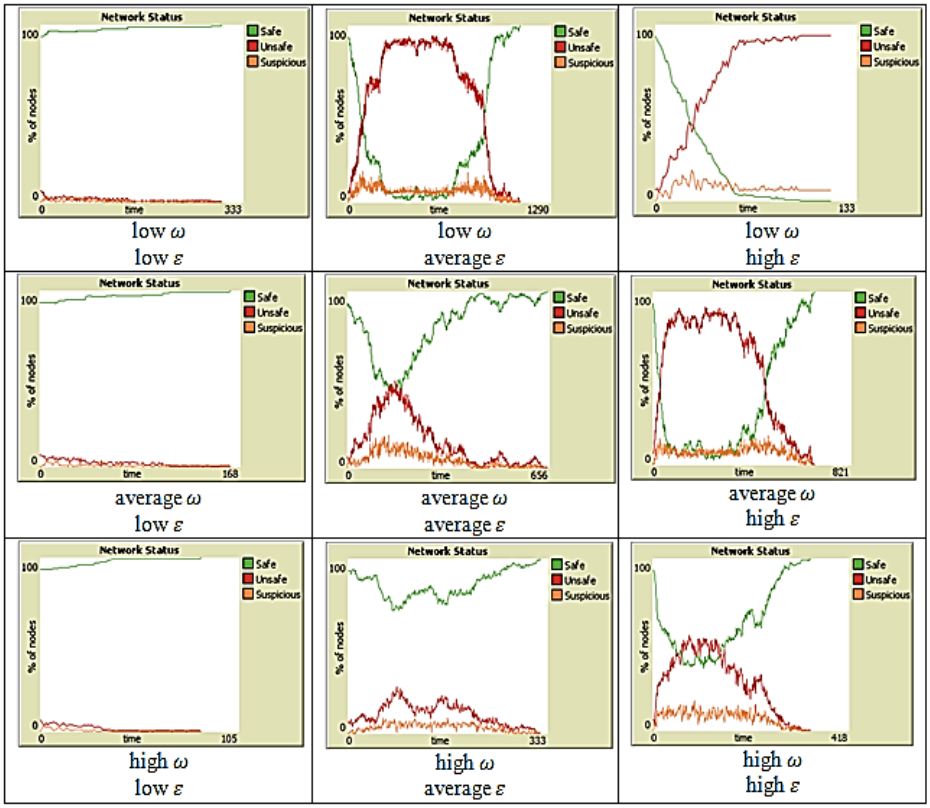

Figure 1: Network Status considering to intrusion response probability percentage (ω) and infection spread percentage (ε)

As it has been mentioned in section 4, increasing the level of agents knowledge causes an enhancement in convergence rate for robust status. Fig. 5 shows the convergence rate of network status to robust status considering ω and ε. With ω enhancement, the convergence rate to robustness is increased and when ε is low, this rate have a significant growth.

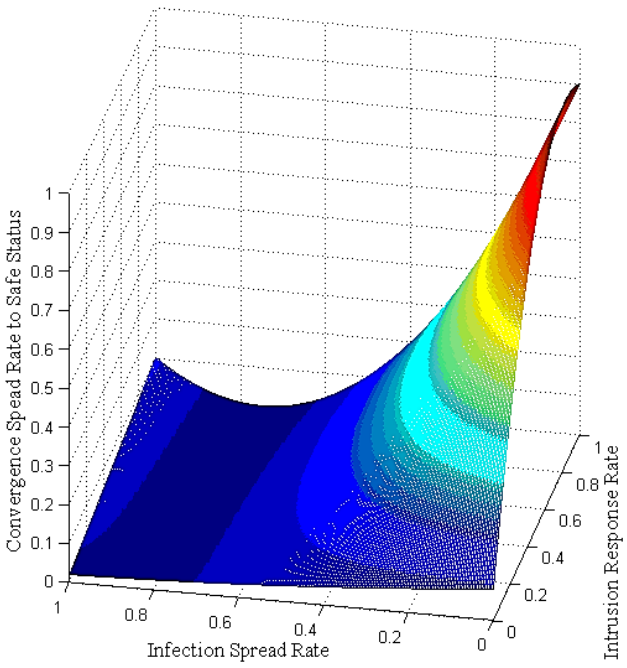

Figure 2: The convergence rate of network status to robust considering to intrusion response probability percentage (ω) and infection spread percentage (ε)

VI. CONCLUSIONS

In this paper, an autonomous multi-agent intrusion detection system has been presented inspired by immune system operation. The first level of intrusion detection is based on the anomaly detection from three views; hardware, software and user. In order to detect an anomaly, systems current behavior is compared with the previous behaviors and the probability percentage of the system failure based on this anomaly is calculated. When this amount is not equal to zero, the node is known as an unsafe node. Therefore, it tries to communicate with one of neighbors nodes' agents to handle this anomaly. In second level of intrusion detection, the detection is based on the intrusion signature. In this detection process, if agents of the neighbor node find the node condition more unsafe than it's been predicted in first level, they will produce an Ab agent and tries to deal with the anomaly (now, it is known as an intrusion). It is important to note that using the detection mechanism in second level for prediction of node unsafe behavior needs clustering methods like Bayesian networks that this paper has not mentioned it.

In addition to the proposed system, a simulation model has been also presented in this paper. The results of this model shows that the presence of immature agents with less knowledge could decrease the convergence rate to the robust status of the network, and if the infection spreads rate is high then the network loses its robustness permanently.

VII. REFERENCES

[1] U. A. Sandhu, A. Haider, S. Naseer, O. U. Ateeb, "A Survey of Intrusion Detection & Prevention Techniques", 16th Singapore, International Conference on Information Communication and Management, 2011, pp. 66-71.

[2] M. Roesch, "Snort – Lightweight Intrusion Detection for Networks", 13th Washington, USENIX conference on System administration, 1999, pp. 229-238.

[3] L. Panait, S. Luke, "Cooperative Multi-Agent Learning: The State of the Art". *Autonomous Agents and Multi-Agent Systems*, 11 (3): pp. 387-434, November 2005. doi:10.1007/s10458-005-2631-2

[4] L. N. de Castro, F. J. Von Zuben, "Learning and Optimization using the Clonal Selection Principles". *IEEE Transactions on Evolutionary Computation*, 6 (3): pp. 239-251, June 2002. doi:10.1109/TEVC.2002.1011539

[5] L. N. de Castro, F. J. Von Zuben, "aiNet: an Artificial Immune Network for Data Analysis". In: Abbass, H.A., Sarker, R.A., Newton, C.S. (Eds.), Data Mining: A Heuristic Approach, pp. 231-259, Idea Group Publishing, USA, 2001.

[6] P. K. Harmer, P. D. Williams, G. H. Gunsch, G. B. Lamont, "A Artificial Immune System Architecture for Computer Security Applications". *IEEE Transactions on Evolutionary Computation*, 6 (3): pp. 252-280, June 2002. doi:10.1109/TEVC.2002.1011540

[7] G. Funk, A. Barbour, H. Hengartner, U. Kalinke, "Mathematical Model of a Virus Neutralizing Immunglobulin Response". *Journal of theoretical biology*, 195 (1): pp. 41-52, November 1998. doi:10.1006/jtbi.1998.0779

[8] S. Forrest, C. Beauchemin, "Computer Immunology". *Immunological Reviews*, 216 (1): pp. 176-197, April 2007. doi:10.1111/j.1600-065X.2007.00499.x

[9] D. E. Kirschner, G. F. Webb, "A Mathematical Model of Combined Drug Therapy of HIV Infection". *Journal of Theoretical Medicine*, 1 (1): pp. 25-34, 1997. doi:10.1080/10273669708833004

[10] R. J. DeBoer, P. Hogeweg, H. F. J. Dullens, "Macrophage T Lymphocyte Interactions in the Anti-Tumor Immune Response: A Mathematical Model". *Journal of Immunology*, 134 (1): pp. 2748-2758, 1985.

[11] S. Bandini, "Hyper-Cellular Automata for the Simulation of Complex Biological Systems: a Model for the Immune System". *International Journal of Applied Science and Computation*, 3: pp. 1, 1996.

[12] N. Fachada, "SimulIm: an Application for the Modelling and Simulation of Complex Systems, Using the Immune System as an Example". Graduation project

report, Higher Technical Institute, Technical University of Lisbon, 2005.

[13] A. Emerson, E. Rossi, "ImmunoGrid - The Virtual Human Immune System Project". *Stud Health Technol Inform*,126: pp. 87-92, 2007.

[14] C. Bianca, M. Pennisi, "Immune system modelling by top-down and bottom-up approaches". *International Mathematical Forum*, 7 (3): pp. 109-128, 2012.

[15] C. M. Ou, "Host-based Intrusion Detection Dystems Adapted from Agent-based Artificial Immune Systems". *Neurocomputing*, 88: pp. 78-86, July 2012. doi:10.1016/j.neucom.2011.07.031

[16] R. B. Machado, A. Boukerche, J. B. M. Sobral, K. R. L. Juc´a, M. S. M. A. Notare, "A Hybrid Artificial Immune and Mobile Agent Intrusion Detection Based Model for Computer Network Operations", 19th Colorado, International Parallel and Distributed Symposium Processing (IPDPS'05), 2005, pp. 191a. doi:10.1109/IPDPS.2005.33

[17] A. Boukerche, R. B. Machado, K. R. L. Juca´, J. B. M. Sobral, M. S. M. A. Notare, "An Agent based and Biological Inspired Real-Time Intrusion Detection and Security Model for Computer Network Operations". *Computer Communications*, 30 (13): pp. 2649-2660, September 2007. doi:10.1016/j.comcom.2007.03.008

[18] A. Byrski, M. Carvalho, "Agent-Based Immunological Intrusion Detection System for Mobile Ad-Hoc Networks", 8th Poland, International Conference on Computational, 2008, pp. 584-593. doi:10.1007/978-3-540-69389-5_66

[19] A. Herrero, E. Corchado, M. A. Pellicer, A. Abraham, "MOVIH-IDS: A Mobile-Visualization Hybrid Intrusion Detection System". *Neurocomputing*, 72 (13-15): pp. 2775-2784, August 2009. doi:10.1016/j.neucom.2008.12.033

[20] K. Murphy, "Janeway's Immunobiology". Garland Science, 2012.

[21] G. D. M. Serugendo, M. P. Gleizes, A. Karageorgos, "Self-Organization in Multi-Agent Systems". *The Knowledge Engineering Review*, 20 (2): pp. 165-189, June 2005. doi:10.1017/S0269888905000494

[22] C. M. Macal, M. J. North, "Tutorial on Agent Based Modeling and Simulation PART 2: How to Model with Agents", California, Winter Simulation Conference, 2006, pp. 73-83. doi:10.1109/WSC.2006.323040

How to cite

Elnaz B. Noeparast, Reza Ravanmehr, "A Two-Level Autonomous Intrusion Detection Model Inspired by the Immune System". *International Journal of Research in Computer Science*, 4 (1): pp. 11-17, January 2014. doi: 10.7815/ijorcs.41.2014.076