ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 2 February , 2014 Page No. 3822-3829

An Immune Inspired Behavior-based Multi-Agent Model for Detecting Network Clients' Misbehavior

Elnaz B. Noeparast[1], Reza Ravanmehr[2], Ramin Nasiri[3]

[1]Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
e-b-noeparast@iauctb.ac.ir

[2]Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
r.ravanmehr@iauctb.ac.ir

[3]Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
r_nasiri@iauctb.ac.ir

Abstract: ***Most of the intrusion detection systems are unable to detect behavior-based intrusions such as Stuxnet, because of their absolute view of the intrusion. There are some legitimate behaviors which their subsequences cause intrusions. In this paper, a multi-agent model inspired by the human immune system has been proposed whose autonomous agents have a conditional view towards intrusion concept. The first level of the intrusion detection in this model has been implemented in clients' side on the anomaly detection. Furthermore, by agent migration to the server, the final detection about the intrusion is fulfilled by server's agents in second level. In this level, an intrusion probability is measured in a Bayesian network based on the subsequence of functions and system calls which has been invoked in the client. This value shows the occurrence probability of this subsequence in an intrusion. Therefore, the false negative error probability will be decreased.***

Keywords: Intrusion Detection, Multi-Agent Systems, Immune System, Autonomous Computing

1. Introduction

Nowadays, most computers use network communicational technology to access utilities such as information exchange as well as providing and receiving services. These facilities hold many advantages for actual and legal users, at the same time, they make computer system vulnerable. Some of these vulnerabilities are; probability of unauthorized access to some parts of the network, failure to provide and receive services, damaging or modifying information

resources and sending them to an outside computer via outsiders' intrusion or malware spreading through the network. In order to handle these threats, intrusion detection systems are used whose main goal is detecting unauthorized use and discovering abusive behavior[1].

Traditional intrusion detection systems work based on malicious behavior patterns. These systems can only react to the known intrusive activities, therefore updating would be necessary[2]. Also there is another manner known as anomaly detection which, detects traffic violations and abnormalities within systems and networks by extracting system behavioral model parameters and mining integrated descriptive statics from these parameters sets[3]. Although, the computational complexity of processing is capable of analyzing a huge amount of data, [4] in real conditions causes some limitations for using this manner. Additional to these problems, mentioned intrusion detection systems detect vulnerabilities via a central element that is an inefficient method for current systems with huge amount of communications and activities.

The artificial immune system is a new computational pattern based on natural immune system processes[5]. This pattern could be an appropriate choice for solving real world complex problems such as anomaly detection [6],[7] and computer system security [8], because the natural immune system is a distributed system and has the ability to learn, memorize and distinguish itself from non-self[9].

This paper organizes as follow: within the next section, a number of relevant works and literatures of the immune system based on intrusion detection systems will be reviewed. Afterwards, an autonomous system including agents and their components will be presented based on this process in section 3. This section provides detailed description of the proposed model for interaction between agents and the related activity diagram. In section 4, the simulation results for the proposed system will be shown as well as evaluation of the convergence rate of network status to robust in different conditions. Finally, concluded remarks and future works would be outlined.

2. Research Literature

2.1. Related Work

There are many researches in the field of intrusion detection inspired by the immune system operation. One of these models is proposed in [1] 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. [10]. 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. [11] alsodeveloped a real-time host-based intrusion detection model for anomaly detection using mobile agents, inspired by the human immune system. Byrski and Carvalho [12] proposed agent-based intrusion detection approach in MANETs, artificial immune systems for anomaly detection,

independent of specific routing protocols and services. Moreover, Herrero and et al. [13] introduced an unsupervised connectionist multi agent intrusion detection system named MOVIH-IDS.

2.2.The Immune System

The body handle infection element with the immune system (IS),which contains lymphocytes and Antigen presenting cells (APC). In this system, part of immunity has been carried out through Antigen-Antibody system whose elements are dendritic cells (DC), B type lymphocytes and T-Helper type lymphocytes.

When an microbial intrusion (bacteria and viruses) happens in a tissue, DCs detect these microbes via their receptors known as Toll-like receptors (TLR)[14]. These cells discover not only the specifications of all intruders' classes, but also structural specifications of each class. Therefore an invader could not escape from DC's analysis by making mutation in its class.

After gathering microbial antigen, DCs immigrate to the nearest lymph node through the lymphatic system. Lymph nodes alike a kidney have virgin and active lymphocytes (T-Helper and B type). DCs present the collected antigens to these lymphocytes when they come to these nodes. [14] B-lymphocytes stated in lymph nodes, after producing in bone marrow are named virgin B lymphocytes. Since these cells have only one pattern, they are able to detect one type of antigen each. This antigen is called the cognate antigen. When a cognate antigen is presented to the B lymphocyte, this cell rearranges its Antibody class by receiving co-stimulatory signal from a T-Helper lymphocyte. This rearrangement helps the B lymphocyte to mutate its antibodies and producing antibodies with the most affinity to the cognate antigen. Afterwards some of the B lymphocytes convert to plasma cells via the signals they receive from T-Helper lymphocytes which secrets their antibody in the environment (the lymph node) and others transform to memory B lymphocytes in order to keep the memory of an intrusion [14].

T-Helper lymphocytes come to lymph nodes after training in thymus, then alike B lymphocytes is activate by meeting their cognate antigen. As it has been mentioned before, they help B lymphocytes to produce antibody. The secreted antibody is released in the blood or interstitial fluid and is conveyed to the battle arena through them. Subsequently, antibodies bind to their cognate antigen in the battle area and destroy the invader[14],[15].

2.3.The Artificial Immune System

The artificial immune system (AIS) is an Evolutionary algorithm inspired by immunology science and the human's body defensive system functionality. The AIS tries to provide reliable and secure systems via modeling the immunological processes. This system has four fundamental techniques (1) negative selection algorithm, (2) immune network algorithms, (3) Clonal selection algorithm (4) and Dendritic cell algorithms and danger theory[16]. However using only these techniques could not represent immunological elements collaboration for implementing tasks in real world. Therefore multi agent systems could be a supplement for the AIS because of their collaboration, communication [1] and distribution ability. These systems are also a suitable solution for distributed intrusion detection systems establishment in networks.

2.4.The Agent-Based Artificial Immune System

A multi agent system (MAS) includes a set of agents which of each have a certain amount of autonomy in their activity domain. The functional resultant of these agents represents total ability of the MAS, which is calculated based on the consequent of each agent autonomic computing [17],[18]. Therefore by using autonomic computing

specifications of the MAS in the AIS, a multi agent AIS could be modeled whose agents are designed based on the autonomic computing architecture. In this system, each agent can mimic some parts of immune system functionalities and have four main phases to achieve self-adaption specific [19],[20]. In the monitoring phase, the agent is aware of internal and external environment conditions and interacts with other agents based on it. The internal environment includes a set of self agents and the external environment includes a set of non-self agents which their existence in internal environment is known as an intrusion. Distinguish between self and non-self as well as the environment condition analysis is done in analysis phase. In plan phase, the agent plans a behavior set, which has the most compatibility with the environment conditions. Flexible behavior makes the agent able to operate in heterogeneous environments and all of platforms. In the execution phase, the agent expresses the planed behavior as a react to the environment condition. The knowledge based in the agent architecture adds learning and information exchange abilities to the agent specifications.

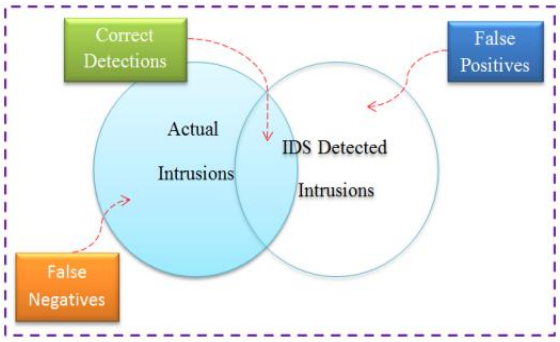

Figure 1: False Positive and False Negative Errors [25]

2.5.Non-Self Detection

Every effort from inside or outside [21] which menaces integrity, confidentiality and availability of resources is known as an intrusion [22] and intrusion detection systems (IDS) detect and handle these intrusions. The main objective of these systems is unauthorized access, misuse and computer violation as well as network resources. IDS uses three fundamental abilities to achieve this goal including; monitoring (evaluation), analysis (detection) and response (reporting) [23],[24].

However there is a possibility of two errors occurrence in these systems; false negative error (FNE) and false positive error (FPE). As it has been shown in Figure 1, if a transaction is an actual intrusion in execution time but IDS knows it as a normal transaction, a FNE is happened. In contrast, a FPE is happened when a normal behavior is known as abnormal.

In AIS, the intrusion is defined as a non-self intrusion to the internal environment. In these systems, if a self is considered as a non-self, a FPE has occurred. On the other side, if a non-self is approved as a self, a FNE has been happened.

3. The Proposed Model

3.1.The Problem Definition

Most of IDSs are inefficient for detecting industrial sabotage and spying malwares such as Stuxnet and kernel rootkits. These malwares divide their behavior or operations to functions and system calls, which are normal from the IDS's system view and causes FNEs. In contrast, if IDSs are strict with systems behavior, the legitimate software and user's operation could be detected as an intrusion, which causes the FPE rate increasing. Also, IDSs need updating and depend on a central element to receive new signatures. Even if an IDS is updated continuously, there still might be a FNE. Polymorph viruses change their structures each time of the proliferation and could not be detected via their previous signature.

The IS as an evolutional system deals with the same problems. There are some viruses such as flu virus, which changes their morph when they are transmitted from a human body to another. Therefore the IS faces a new kind of intrusion.

Also, there are some smart viruses which damage the MHC Class I molecule-expressing ability in cells. These molecules are on the surface of most cells and act like billboards to show what is happening inside the cell in order to immune them. If there is no sign of these molecules on the cell's surface, the IS could not detect the intrusion. Also, IS should deal with cancer cells and detect abnormal behavior of these cells from normal ones [14],[26].

The proposed model of this article is inspired by mentioned IS operations and the non-selfconcept is a behavior which causes system failure. Intended for evaluating a non-self behavior, system behavior is monitored from three points of view (Hardware, Software, and User's view). A statistic average is measured for each 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) is considered whose value is between a minimum and its consuming estimate. 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. Then the standard deviation value percentage is applied according to the Eq. (1), and the system's failure probability percentage (δ) is calculated.

$$\delta = \frac{C_H + C_S + C_U}{3 \times 100} \quad (1)$$

where C_H, C_S and C_U parameter is deviation percent for hardware, software and user behavior orderly.

3.2. The Software Architecture

The proposed model uses four kinds of agents, which are distributed through a client server network and collaborate with each other to detect and handle intrusions. These agents are DC agents, TH agents, B agents and Ab agents, which their autonomic computing architecture and communications has been presented in Figure 2.

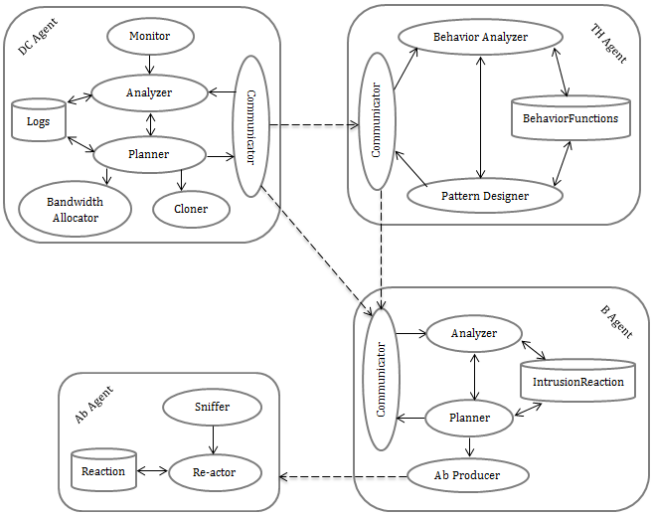

Figure 2: The proposed system (agents and their components)

3.2.1.DC Agent

DC agent acts like a DC cell and carries out the internal environment cognition of the clients' node and analysis of their conditions from three points of views (hardware, software, and user's view) which has been mentioned before. This agent records behaviors, system statuses and operational logs of the client's system to be able to track them. Also it compares the current situation and the average of previous ones through three mentioned views. Afterwards, the standard deviation value percentage is applied according to the formula as it has been mentioned in part 3-1. Subsequently, DC agent clones an Emigrant DC agent, which has the same structure. This agent takes the client's system's snapshot, which contains all occurred actions in the system from three viewpoints and generates an identifier which is unique through the network. This identifier is named as $ID_{Anomaly}$. Afterwards, it saves $ID_{Anomaly}$, δ and the snapshot in its knowledgebase and migrates it to the server node where TH and B agents are stated. In the server node, the Emigrant DC agent presents

$ID_{Anomaly}$, δ and the snapshot to TH and B agents as an intrusion detection and handling request from the client node. DC agent's activity diagram has been shown in Figure 2.

This agent also sends a dedicated bandwidth allocation request to its direct and indirect neighbors, which locates between self-node and server node. This bandwidth leads Ab agent towards client's node. Its details has been explained in part 3-4. Ab agent is produced based on snapshot information by B agent and handles an intrusion, which has been mentioned in part 3-2-4.

As it has been shown in Figure 1, the autonomous architecture of DC agent has six modules and a knowledgebase as below;

- Communicator: This module is responsible for sending the snapshot to TH, B and Ab agents.
- Monitor: This module monitors the system status from hardware, software and user's views.
- Analyzer: This module is responsible for analyzing the system status regularly and recording operational log in Logs knowledgebase.
- Planner: This module compares current system condition with the previous situation and calculates δ.
- Cloner: this module clones an Emigrant DC agent whose structure is the same as DC agent's structure.
- Bandwidth Allocator: This module allocates a dedicated bandwidth between self-node and server node by sending a dedicated bandwidth allocation request to neighbors.
- Logs: It's a knowledgebase including operation's logs and system statuses.

3.2.2.TH Agent

This agent analyzes the snapshot's information, which is received from an Emigrant DC agent. Then it extracts occurred behaviors and splits them to the pattern of functions and system calls known as signatures and sends these signatures to B agent. The activity diagram of TH agent is presented in Fi No. 3. This agent includes three modules and a knowledgebase as shown 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 the Colored Petri Nets (CPN) [27] column of BehaviorFunction knowledgebase. In this case, each behavior, which could pass the token through one of CPNs paths to the endhas been sent to Pattern Designer module.
- Pattern Designer: Since there is different operations and system calls which might express the same behavior, this module determines the signatures which are subsequences of operations and system and may cause the received behavior from Behavior Analyzer.
- BehaviorFunction: This knowledgebase has a Column Family data model including anomaly behaviors as well as the signatures caused those behaviors. The behaviors' information is CPNs located in key columns of knowledgebase table also, functions and system calls which may cause a behavior located in SuperColumns.

3.2.3.B Agent

B agent's operation is based on B cell in the immune system; it carries out the intrusion detection and extracts suitable behaviors in order to respond to an intrusion. This agent receives the signatures from TH agent and $ID_{Anomaly}$, δ and

the snapshot from DC agent. Then it compares these signatures with the functions patterns and system calls in its knowledgebase. If there is a pattern matched with a signature, B agent extracts the intrusion probability percent (γ) of that pattern and compares it with δ. If the result of γ is bigger than δ, it means the anomaly is actually an intrusion. Hence, B agent produces an Ab agent and injects the suitable behaviors, the snapshot and $\mathrm{ID}_{\mathrm{Anomaly}}$ into Ab agent's knowledgebase. The activity diagram of B agent has been shown in Figure 3. This agent's architecture has four modules and a knowledgebase as shown as below;

- Communicator: This module receives$\mathrm{ID}_{\mathrm{Anomaly}}$, δ and the snapshots from DC agent along with signatures from TH agent.
- Analyzer: This module analyzes received signatures and compares them to its knowledgebase patterns, then determines the γ. The comparison results could be one of these two cases: 1) the signature is matched to one of the intrusion pattern network paths, 2) the signature is similar to one or more of these paths. In case (1), the γ value is measured before, but in case (2) Analyzer module needs to calculate it. Intended for calculating the γ value in case (2), Analyzer module produces new patterns based on its knowledgebase patterns and compares them with the signature. By finding a matched pattern, Analyzer module calculates the γ value based on the measured γ values of its patterns and sends the matched pattern with its γ value to the Planner module.
- Planner: this module is responsible for detecting the accuracy of an intrusion and decision on producing Ab agents. If the result 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 B agent's knowledgebase 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. Afterwards, it sends $\mathrm{ID}_{\mathrm{Anomaly}}$, the snapshot and the behaviors in order to respond to intrusion to Ab Producer module.
- Ab Producer: This module produces an Ab agent and injects $\mathrm{ID}_{\mathrm{Anomaly}}$, the snapshot and the response behaviors in Ab agent's knowledgebase.
- Intrusion Reaction knowledgebase: This knowledgebase includes two sets of knowledge. The first one is intrusion knowledge includes a function Bayesian network and system calls subsequences called the Intrusion Pattern Network, which may help an intrusion. In this network the occurrence probability of each functions and system calls is determined and the probability of their subsequences [28] is calculated as a γ for each pattern. The classification of the Intrusion Pattern Network could be done in different ways such as naïve bayes[29], Tan[30], BAN[31] and ABC-Miner[32] which is out of the scope in this paper. The second set includes the intrusion response behavior set named Response Network, which is corresponding to the Intrusion Pattern Network. Hence, the intrusion detection and express a behavior is a mapping from the Response Network to the Intrusion Pattern Network. This mapping could be very simple and include only one specific behavior or it might be very complex and contain a network of conditional behaviors

each one to match to one of the Intrusion Pattern Network's paths.

3.2.4.Ab Agent

Ab agent has similar characteristics to the immune system antibody and it is produced 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 walks through the network randomly to find another node with anomaly and similar events log with its recorded snapshot. This agent has two modules and a knowledgebase as below;

- Sniffer: This module is responsible for finding a dedicated path towards an infected node with the same $\mathrm{ID}_{\mathrm{Anomaly}}$ as the $\mathrm{ID}_{\mathrm{Anomaly}}$ in its knowledgebase or an infected node with the similar events log alike the snapshot in its knowledgebase.
- Re-actor: This module responds to intrusions and removes their effects.
- Reaction knowledgebase: This knowledgebase includes intrusion responses, snapshot and $\mathrm{ID}_{\mathrm{Anomaly}}$.

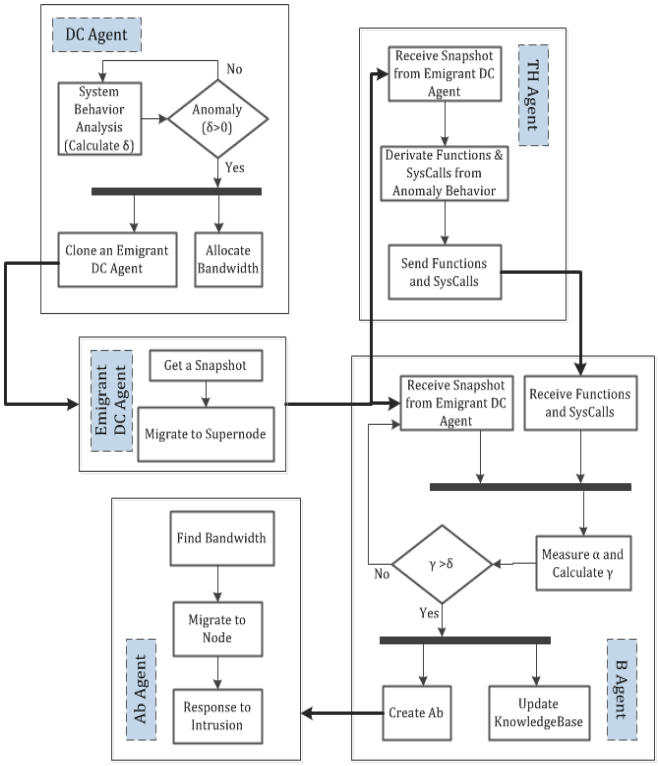

Figure 3: Agents' activity diagram

3.3.The Dynamic communication infrastructure

In multi agent AISs such as [1] whose responded agents are in the peer or central node, there might be few changes in the infected system conditions because of the time interval among intrusion detection and intrusion response. This time interval sometimes makes the intrusion handling mechanism unprofitable and sometimes causes problems for the infected system. Therefore a new mechanism is needed to keep response agents aware of the infected system conditions during the time interval.

Since millions years ago, insects such as ants could survive in different environments and climate conditions unlike dinosaurs. The secret of this ecological success is behind the fact that ants live as a colony. Ants colony is a distributed self-organization whose objects act through a paradigm named stigmergy. This paradigm is an efficient and asynchronous communication mechanism, which changes the environment by a chemical volatile substance named pheromone. This substance evaporates gradually after propagation via ants. The continuity secretion of this substance in a path creates a pheromone

path, which attracts other ants and causes a self-catalytic behavior. Also the evaporation mechanism provides dynamic interaction ability between the colony and the environment, hence an optimal behavior resultant of colony to environment conditions occurs.

In the proposed model, the communication of DC agents and Ab Agents is designed based on stigmergy paradigm. After a DC agent measures the δ and its substance cloned by its migration to the supernode, DC agent allocates a dedicated bandwidth for Ab agent through its node's neighbors stated between the infected node and supernode. Using this mechanism, DC agent tries to attract the Ab agent towards infected node and Ab agent, which prefers paths with less traffic movements to the infected node side.

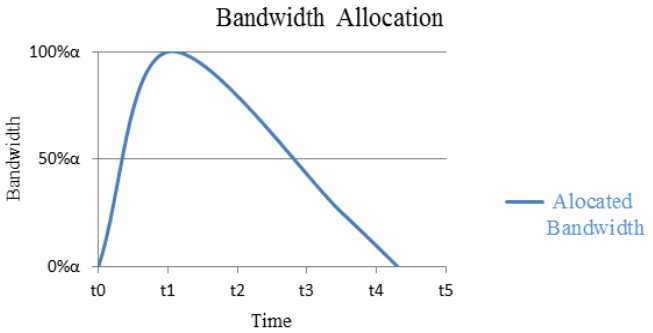

Figure 4: Dedicated bandwidth allocation for an Ab agent during a period of time

The dedicated bandwidth is allocated based on the request sent to neighbor nodes by DC agent. This request has the pheromone information as mentioned bellow;

1. Pheromone Identifier ($ID_{Fermonea}$): This Identifier is unique in the whole network and is the identifier of a dedicated bandwidth.
2. Sensitivity (S): This value is equal to δ and has a direct relevance with the allocated bandwidth. In other words, the bigger value of sensitivity makes nodes to allocate more bandwidth, therefore the Ab agents could transfer to the infected node faster.
3. Allocated Bandwidth for a Request (α): The pheromone information propagation in the network makes a dedicated path between that infected node and supernode which evaporates after gaining its most value. This process has been shown in Figure 4.

When an Ab agent enters to network, sniffs a dedicated bandwidth whose $ID_{Fermonea}$ is the same as the $ID_{Anomaly}$ in its knowledgebase received form the Emigrant DC agent indirectly. By finding the allocated path in each node, it goes to infected node directly. In this case, because of the dedicated path existence, Ab agent could be certain that its presence is needed in the infected node. If there is not any dedicated bandwidth, it means infected node's condition is updated. This update could happened under two conditions: 1) The infected node has come back to normal condition, so there is no need to Ab agents, 2) There is a new abnormal condition which needs another Ab agent. Therefore the Ab agent starts a stochastic action and walks through the network randomly to find another infected node with the same conditions.

4. Simulation Results

In order to simulate the proposed model, NetLogohas been used which is an agent-based simulation environment [33]. As it is shown in Figure 5, there is a network which their client nodes has a direct or indirect connection with a server node (with gray color). When an anomaly is occurred, based on a DC agent in a client node, the node's color is changed from green to red. Afterwards the DC agent sends a dedicated bandwidth allocation request to its neighbors between the self-node and server node. Each allocated bandwidth is shown with an orange color path between two nodes, also the thickness is corresponding to δ and its Sensitivity value is presented beside it in a bracket. Hence, if a δ

value is bigger than the other, the allocation bandwidth through the network is more (the dedicated path between nodes is thicker) and Ab agent transferring has more priority for this path. After the bandwidth allocation and Emigrant DC agent migration, if B and TH agents in server node could produce an Ab agent before the pheromone evaporation (dedicated path), Ab agent would be able to reach to the infected node directly via this path (No. 1 situation). Otherwise, the path is evaporated and node color remains red (No. 2 situation).

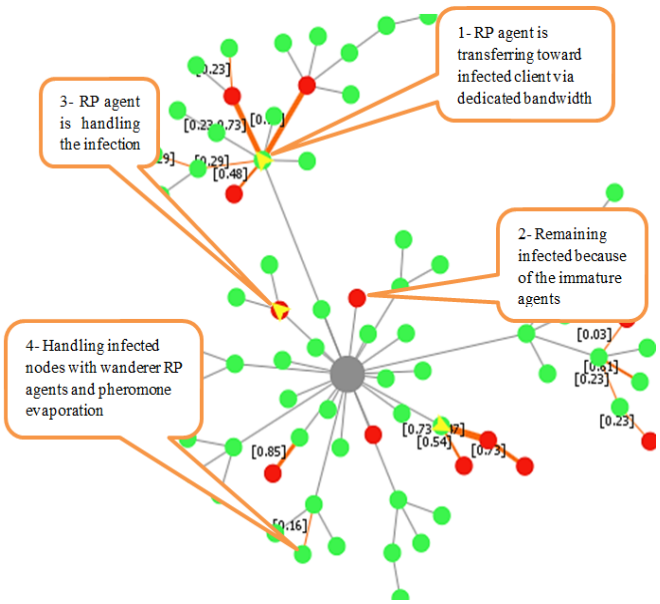

Figure 5: The simulation environment

After the Ab agent arrived to the infected node, this agent handles the infection (intrusion and its side effects) based on the snapshot and response behavior in its knowledgebase (No. 3 situation). After handling the infection, Ab agent would transform to a wanderer Ab agent and would walk through the network randomly to find another infected node with anomaly and similar events log with its snapshot. In this case, it is possible that a node infection is removed when its DC agent is propagating the pheromone through the network, so the DC agent stops sending requests to neighbors to allocate a dedicated bandwidth (No. 4 situation).

As it has been shown in situation No. 1 of Figure6, When B and TH agents are immature, the intrusion detection and handling rate is low, so the count of infected nodes increases through the network by infection spread and the intrusion detection system could not reduce the infected nodes count. Since the server node's agents have learning ability as well as updating their information based on environmental conditions, they would be able to detect more infections over time. Accordingly, there is an impalpable decrease in the count of infected nodes in situation No. 2 despite the growth of infection spread. This reduction amount is more sensible in situation No. 3 regarding the agents' awareness enhancement. But this awareness is not sufficient to handle all infected nodes, therefore an overall increase in the count of infected nodes is likely to occur. In situation No. 4 there is a stair decline additional to primary decrease, because of wanderer Ab agents, which handle some of infected nodes. These agents distribution throughout the network and B and TH agent with more maturity cause an impressive reduction in the count of infected nodes, which reaches the lower value than primary count of infected nodes. This case is marked in situation No. 5 and shows that the level of intrusion detection system maturity makes it capable of detect and handle most of infection occurred up to this stage. Situation No. 6 is marked to show the absolute maturity of B and TH agents. Also the network status is converged to robust. In this stage, even if an unknown infection happens in some of client's nodes, the intrusion detection system is able to handle all infections and decreases the count of infected nodes to none. In this situation, wanderer Ab agent's multiplicity throughout the network causes the Real-Time nodes' infection handling.

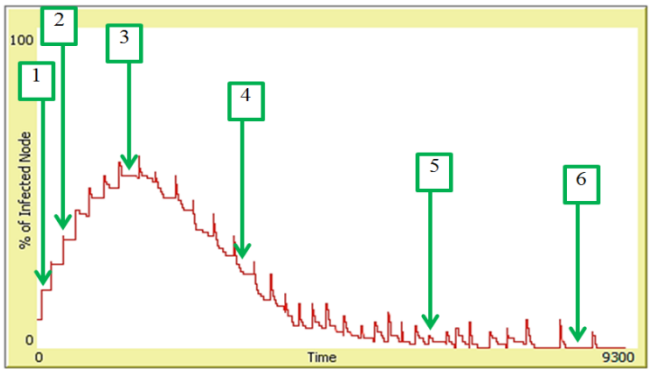

Figure 6: The convergence of network status to robust considering to new and unknown intrusion existance

Taken into consideration, mentioned details and overall view to the operation of the proposed intrusion detection system in Figure 6 is comprehended that B and TH agents are immature at the start point, consequently the count of infected nodes would increase significantly as a result of low rate of detection and handling. However this count and the server load overhead are decreased over time as agents' learning ability and wanderer Ab agents' existence and the network status would be converged to a robust status in a period of time. This robustness is maintained during the time, even if there are new and unknown intrusions happening.

5. Conclusion

In this paper, an autonomous multi-agent intrusion detection system has been proposed inspired by humans' body immune system operation. The first level of intrusion detection is based on the anomaly detection from three point of views; hardware, software and users'. Afterward the standard deviation value percentage has been calculated based on comparing current situation with the average of previous ones throughout three views. If this amount is not equal to zero, an agent has been cloned from the node's agent. Then it takes a snapshot from system log and migrates to the server to start second level of intrusion detection. After absolute intrusion detection, the server's agents create an intrusion-handling agent with the most appropriate handling response known as Ab agent. During this period of time, the infected client's agent provides a dedicated path inspired by Ant's pheromone propagation leading the produced Ab agent towards the client. This manner establishes an asynchronous connection between client and Ab agent, which makes Ab agent aware of clients' status. If there is no need for Ab agents' existence in that client, this agent would try to find another client with similar condition and decreases the overhead ofproducing similar Ab agents by the server.

References

[1] C.M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," Neurocomputing, 88, pp. 78-86, 2012.

[2] Sharada K. A., Hemant, Prashanth, and Vijay kumar S., "A Model Proposed for Reducing the False Positive Alarm Rate Using the feature of Event Correlation," International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 2 (8), pp. 103-108, 2012.

[3] I. Khalkhali, R. Azmi, M. Azimpour-Kivi, and M., Khansari, "Host-based Web Anomaly Intrusion Detection System, an Artificial Immune System Approach," International Journal of Computer Science Issues (IJCSI), 8 (5), pp. 14-24, 2011.

[4] M. Ostaszewski, F. Seredynski, and P. Bouvry, "Coevolutionary-based Mechanisms for Network Anomaly Detection," Journal of Mathematical Modelling and Algorithms, 6(3), pp. 411-431, 2007.

[5] S.M. Garret, "How do we evaluate artificial immune systems," Evolutionary Computation, 13 (2), pp. 145-178, 2005.

[6] M. Ayara, J. Timmis, R. de Lemos, and S. Forrest, "Immunising automated teller machines," In Proceedings of Springer Int. Conference on Artificial Immune Systems, 3627, pp. 404-417, 2005.

[7] F.A. Gonzalez, and D. Dasgupta, "Anomaly detection using real-valued negative selection", Genetic Programming and Evolvable Machines, 4 (4), pp. 383-403, 2003.

[8] P.K. Harmer, P.D. Williams, G.H. Gunsch, and G.B. Lamont, "An artificial immune system architecture for computer security applications," IEEE Transactions on Evolutionary Computation, 6 (3), pp. 252-280, 2002.

[9] V.S. Aragóna, S.C. Esquivela, and C.A.C. Coellob, "A T-Cell Algorithm for Solving Dynamic Optimization Problems," Information Sciences, 181 (17), pp. 3614-3637, 2011.

[10] R.B. Machado, A. Boukerche, J.B.M. Sobral, K.R.L. Juc´a, and M.S.M.A. Notare, "A Hybrid Artificial Immune and Mobile Agent Intrusion Detection Based Model for Computer Network

Operations," In Proceedings of IEEE International Parallel and Distributed Processing Symposium, pp. 191a, 2005.
[11] A. Boukerche, R.B. Machado, K.R.L. Juca´, J.B.M. Sobral, and 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, 2007.
[12] A. Byrski, and M. Carvalho, "Agent-Based Immunological Intrusion Detection System for Mobile Ad-Hoc Networks," In Springer International Conference on Computational, Vol. 5103, pp. 584-593, 2008.
[13] A. Herrero, E. Corchado, M.A. Pellicer, and A. Abraham, "MOVIH-IDS: A mobile-visualization hybrid intrusion detection system," Neurocomputing, 72, pp. 2775-2784, 2009.
[14] K. Murphy, Janeway'sImmunobiology, 8th,Garland Science, New York, 2012.
[15] W.E. Paul, Fundamental Immunology, 7th ed., Lippincott Williams & Wilkins, Philadelphia, 2012.
[16] C.C. Kiang, and R. Srinivasan, "An artificial immune system for adaptive fault detection, diagnosis and recovery," International Journal of Advances in Engineering Sciences and Applied Mathematics, 4(1-2), pp. 22-31, 2012.
[17] G.D.M. Serugendo, M.P. Gleizes, and A. Karageorgos, "Self-organization in multi-agent systems," The Knowledge Engineering Review, 20 (2), pp. 165-189, 2005.
[18] K. Ahuja, and P. Ahuja, "A Survey of Methodologies: Component, Aspect and Agent," International Journal of Advanced Research in Computer Science and Software Engineering, 3 (5), pp. 1063-1068, 2013.
[19] D. Garlan, B. Schmerl, and S.W. Cheng, "Software Architecture-Based Self-Adaptation", Autonomic Computing and Networking, Springer, Germany, 2009.
[20] T. Cioara, I. Anghel, I. Salomie, M. Dinsoreanu, G. Copil, and D. Moldovan, "A self-adapting algorithm for context aware systems," In Proceedings of International IEEE Roedunet International Conference, pp. 374-379, 2010.
[21] A.K. Jones, and R.S. Sielken, "Computer System Intrusion Detection: A Survey Technical Report," Univ. of Virginia, Thornton Hall, USA, 2004.
[22] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system," Univ. of New Mexico, Albuquerque, New Mexico, Tech. Rep. CS90-20, 1990.
[23] H. Jadidoleslamy, "A High-Level Architecture for Intrusion Detection on Heterogeneous Wireless Sensor Networks: Hierarchical, Scalable and Dynamic Reconfigurable," International Journal of Wireless Sensor Network (WSN), 3 (7), pp. 241-261, 2011.
[24] P. Prasad, "A Dynamically Reconfigurable Intrusion Detection System," M. Sc. thesis, University of North Carolina State, North Carolina, USA, 2003.
[25] G.A. Fink, B.L. Chappell, T.G. Turner, and K.F. O'Donoghue, "A Metrics-Based Approach to Intrusion Detection System Evaluation for Distributed Real-Time Systems," In Proceedings of IEEE Parallel and Distributed Processing Symposium, pp. 93-100, 2002.
[26] L.M. Sompayrac, How the Immune System Works, 4th, Wiley-Blackwell, Oxford, UK, 2012.
[27] J. Saldhana, and S.M. Shatz, "UML Diagrams to Object Petri Net Models: An Approach for Modeling and Analysis," In Proceedings of International Conference on Soft- ware Engineering and Knowledge Engineering, pp. 103-110, 2000.
[28] G.F. Cooper, and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," Machine Learning, 9 (4), pp. 309-347, 1992.
[29] D. Upadhyaya, and S. Jain, "Hybrid Approach for Network Intrusion Detection System Using K-Medoid Clustering and Naïve Bayes Classification," International Journal of Computer Science Issues (IJCSI), 10 (3), pp. 231-236, 2013.
[30] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," Machine Learning, 29, pp. 131-163, 1997.
[31] J. Cheng, and R. Greiner, "Learning Bayesian Belief Network Classifiers: Algorithms and System", in Proceedings of Springer Biennial Conference of Advances in Artificial Intelligence, pp. 141-151, 2001.
[32] K.M. Salama, and A.A. Freitas, "ABC-Miner: An Ant-Based Bayesian Classification Algorithm," In Proceedings of Springer Int. Conference (ANTS 2012), Swarm Intelligence, pp. 13-24, 2012.
[33] C.M. Macal, and M.J. North, "Tutorial on Agent Based Modeling and Simulation, Part 2: How to Model with Agents," In Proceedings of Winter Simulation Conference, pp. 73-83, 2006.