A Multi-Agent-based Approach to Improve Intrusion Detection Systems False Alarm Ratio by Using Honeypot

Babak Khosravifar
Dept. Comp. Eng. Concordia University, Montreal, Canada
b_khosr@encs.concordia.ca

Maziar Gomrokchi
Dept. Comp. Science. Concordia University, Montreal, Canada
m_gomrok@encs.concordia.ca

Jamal Bentahar
Concordia Ins. for Inf. Sys. Eng. University, Montreal, Canada
bentahar@ciise.concordia.ca

Abstract

In this paper we propose a new architecture, which is composed of distributed cooperative agents to reduce the false alarm ratio of the intrusion detection systems (IDS) in a twofold contribution. The first contribution lies in reducing the false alarm rate of the attack detection in an agent-based architecture by using honeypot network as the closer level of investigation. The connection is retrieved to the original destination in case of false alarm recognition, while the actions are hidden to the user. Such a scheme significantly decreases the alarm rate and provides a higher performance of IDS. The second contribution applies the game theoretic analysis in the sense that the contributing agents are led to perform the best they could in order to achieve their goals. The Shaply value is computed to find the actual contribution of each agent in the coalition he belongs to. The Equilibrium Point is found and consequently the winner coalition is formed. In this paper the architecture of the proposed system is described, a theoretical analysis of agents' behavior is given and its possible extensions are explained.

Keywords. *Intrusion Detection System, Honeypot, Multi-Agent System, Game Theory.*

1. Introduction

Over the past few years, the improvement of *intrusion detection systems* (IDS) has been broadly studied from diverse perspectives. One of the recent approaches to enhance the efficiency of the IDS is to use the idea of distributed artificial agents, known as *multi-agent systems* (MAS) [7], [10], [11]. Applying MAS as the basis for IDS makes detection process decentralized, which is generally a solution for scalability in the sense that it avoids the bottleneck and single point of failure in the systems with real time response.

The aforementioned agent-based architectures mainly have two problems. Firstly the typical proposed frameworks are used to facilitate the intrusion recognition by dividing the tasks into subtasks to be done by agents, while all agents and subtasks are implemented at IDS level [7], [10]. Generally these frameworks lack the detailed discussion in terms of methodology they use to enhance the efficiency of intrusion detection compared to a conventional IDS. Secondly in such systems the distributed agents lack the explicit clarification of their goals, beliefs and desires in a certified manner to enable the frameworks to prove that the multi-layered agent architecture, to some extent, guarantees the approach to the defined goal(s) [7], [11]. In this paper, we propose a framework in which intelligent agents are used as a system component in order to perform spontaneous automated intrusion response. In this framework, agents are characterized to be reactive (in the sense that they perceive the environment and provide responses towards their objectives), proactive (they pursue their defined goals by taking the initiative) and deliberate (they apply the three level structure knowledge of belief, desire and intention so called BDI). Objectively, we advance the conventional use of MAS in IDS by using game theoretical analysis. Like our previous paper [3], we utilize *honeypot* (as bait in the form of a vulnerable system to trap intruders) together with IDS for a more close investigation. The main contribution of this paper is on using cooperative game theory to evaluate our proposed agent-based model. We analyze agents' cooperative behaviors and then formalize this analysis to trace the coalition formation stages and find the optimum coalition (*equilibrium point*).

The rest of the paper is organized as follows. Section 2 presents an architectural model of the proposed framework with the relative components. Section 3 deals with game theoretic approach to discuss the distributed agents' associated actions in a manner that decreases the false alarm ratio of detection. We also provide an expressive example of the discussed issues. In Section 4, we analyze the proposed model's efficiency by comparison with conventional systems. Finally, Section 5 concludes the paper.

2. Proposed Remote Manager Architecture

General Description. In this section we outline the requirements of the proposed infrastructure. The proposed topology in this paper is a cooperative multi-agent system. In order to regulate execution of this MAS and recognize the improvement of intrusion detection, proper methodology

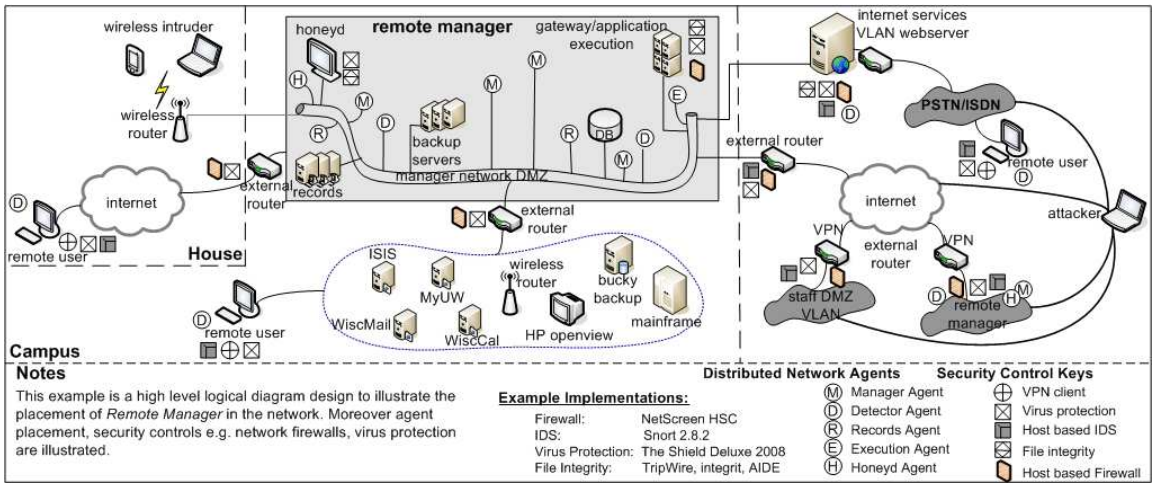

Figure 1. Cooperating MAS infrastructure with remote manager's placement in the network.

must be built to implement the cooperation and communication between the defined agents. The MAS structure is composed of intelligent distributed agents, which are capable of learning and reasoning toward achieving their goals.

The proposed cooperating MAS (illustrated in Figure 1), is set up in distributed sub-nets, consists of different servers and clients, each associated with a host-based IDS. Since the supervised network is composed of a number of LANs, the proposed MAS is configured as three layered agent architecture and each layer consists of different cooperating agents. Agents fall into the following categories: manager agent $M.Ag$, detector agent $D.Ag$, honeyd agent $H.Ag$, records agent $R.Ag$ and execution agent $E.Ag$. The remote manager, as the main part of the topology, associates these agents to the system components. Associated agents communicate with each other and reflect the outcome of the system component to one another. They communicate by exchanging communication messages. A communication message is a tuple $\langle \alpha, \beta, Ag_x, Ag_y, M, t \rangle$ where α indicates whether it is a report, order or fetch communication message, β represents the type of the message (*i.e.* reporting new attack, requesting information, initializing or stop interacting), Ag_x and Ag_y are respectively the sender and receiver of the message, M is the content of the message and finally t is the time at which the message is sent.

Since the remote manager is more complicated than other parts, we magnified its partitioned infrastructure separately in Figure 1. The details cooperative components are in [3]. The remote manager architecture consists of five parts: (1) manager agent $M.Ag$ to make decisions by creating appropriate template for saving relative data called signatures [4]; (2) Records agent $R.Ag$ associated to records as a database, which used to save signatures relevant to intrusions; (3) Honeyd agent $H.Ag$ assigned with honeyd as a pseudo net used to prevent false blocking; (4) Execution agent $E.Ag$ to transform the $M.Ag$'s commands into executable orders; and (5) detector agent $D.Ag$ reflecting the intrusion detection reports of IDS to $M.Ag$.

Manager Agent. When any $D.Ag$ recognizes a suspicious attempt, executes his report plan and sends the corresponding report to $M.Ag$ together with the relative information he obtains from the first interaction. Figure 2 represents the cooperating MAS in a three layer hierarchical tree structure. The $M.Ag$ agent is equipped with a report analysis section, which enables him to check the type of the suspicious attempts. Checking the message content, $M.Ag$ compares the situation risk with the *Agent Risk Threshold*, which is used to warn an attack if interaction exceeds some specifications. In order to be more accurate in checking, $M.Ag$ may query some information from records to be provided by $R.Ag$. In case the risk of attempt does not exceed the threshold, $M.Ag$ orders $D.Ag$ to continue interaction together with the information provided by the $R.Ag$. In contrast, if the risk of the attempt exceeds the threshold, $M.Ag$ in order to cope with the suspicious attack initializes $H.Ag$ to deal with the connection and also $D.Ag$ to stop interacting with the suspicious attacker. $M.Ag$ also requests $E.Ag$ of the router to redirect all the relative connection packets to the specific $H.Ag$ and reroute the responses to the end user. We formalize the summarized communication protocol between the agents (as a if-then rule) as follows:

$$
\begin{aligned}
&\langle Report, N_{attack}, D.Ag, M.Ag, Inf_{intr}, t_0 \rangle \Rightarrow \\
&\quad \langle Fetch, Intr, M.Ag, R.Ag, Inf_{intr}, t_1 \rangle \wedge \\
&\quad \langle Order, Continue, M.Ag, D.Ag, Inf_{R.Ag}, t_2 \rangle \\
&\quad \vee \ \langle Order, Stop, M.Ag, D.Ag, Inf_{R.Ag}, t_2 \rangle \wedge \\
&\qquad \langle Order, redirect, M.Ag, E.Ag, Inf_{intr,H.Ag}, t_2 \rangle \wedge \\
&\qquad \langle Order, Initialize, M.Ag, H.Ag, Inf_{intr}, t_2 \rangle
\end{aligned}
$$

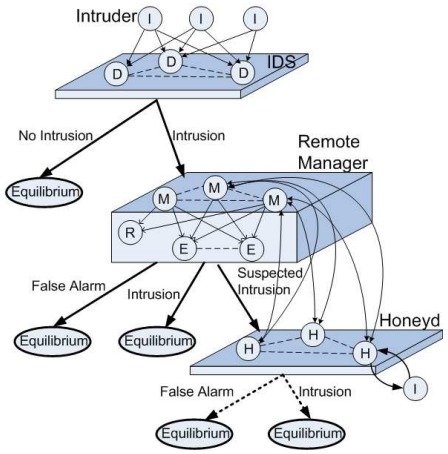

Figure 2. Tree hierarchical structure of MAS.

The $H.Ag$ is a cooperative intelligent agent, which is activated and ruled by $M.Ag$ in the system. $H.Ag$ is equipped with a mechanism capable of learning through introspection and analysis of the intruder behavior and consequently creates the appropriate script for emulating the system. Honeyd can support different services simultaneously, so it starts responding and interacts with the suspected intruder under the control of the $H.Ag$. Honeyd records in another logbook all activities observed while interaction. He also stores all tools used by the intruders in records as a forensic machine. At this time, $M.Ag$ analyzes the interactions taken place between the end user and honeyd which have been reported by the associated $H.Ag$. The analysis is done based on previous knowledge, which can be dynamically updated (by other $M.Ag$s or by administrator) throughout the system functioning. $M.Ag$ is an intelligent agent, therefore he senses the environment and reconfigures the responses. $M.Ag$ also learns from the previous report analysis to enhance his accuracy in intrusion detection. In case of the attack approved, $M.Ag$, concerning the appropriate created signature issued to the records, sends a message to $R.Ag$ followed by a message to $E.Ag$, requests dropping of the packets from that end user. This simply means $M.Ag$, creates the signature upon experience gained from the intrusion attempt of a particular end user which by logging in records, will be used for further likely intrusions of the other users. Moreover $M.Ag$ disconnects the spurious connections, which may lead to loss of resources in the system. In case the attack is disapproved, $M.Ag$ manages to avoid disconnecting the current connection with the end user. To this end, $H.Ag$ is ordered to stop interacting and sends all the logfiles back to the $D.Ag$ that initially reported the attack. Also $E.Ag$ is ordered to redirect the incoming packets to the $D.Ag$ in order to continue interaction. In any case the detection process is not redirected to any other layer, the involved agents pursue the detection process and manage to handle it. This allows for achieving what we call equilibrium points (Figure 2), which will be explained in Section 3. The agent interactions are formalized as following communication messages:

$$
\begin{aligned}
&AttackApproved \Rightarrow \\
&\quad \langle Report, Attack, M.Ag, R.Ag, Inf_{intr}, t_3 \rangle \wedge \\
&\quad \langle Report, Attack, M.Ag, E.Ag, Inf_{intr}, t_3 \rangle \wedge \\
&\quad \langle Order, Stop, M.Ag, H.Ag, Inf_{intr}, t_3 \rangle \wedge \\
&\quad \langle Report, Attack, M.Ag, D.Ag, Inf_{intr}, t_3 \rangle \\
&\quad AttackDisapproved \Rightarrow \\
&\quad \langle Order, Stop, M.Ag, H.Ag, Inf_{intr,D.Ag}, t_3 \rangle \wedge \\
&\quad \langle Order, Redirect, M.Ag, E.Ag, Inf_{intr,D.Ag}, t_3 \rangle \wedge \\
&\quad \langle Order, Continue, M.Ag, D.Ag, Inf_{intr,H.Ag}, t_3 \rangle
\end{aligned}
$$

3. Game Theoretic Analysis

Formalization. In the proposed model, each agent behavior (set of strategies) is a complementary action for other agents in the sense that it affects their decision making process (strategy selection). Without loss of generality, agents adapt themselves to the environment changes, which could be caused by diverse work load on each agent. Therefore, the adaptation to the new state of environment should be an optimum combination of different agents' contributing efforts. In this section, we formulate interactions between different types of agents that use self interested entities in a game theoretic model. Therefore, the system administrator would be able to predict the optimum combination of agents (agent coalition) at a given time considering the load of user interactions. To do so, we model the interaction between different types of agents as a finite dynamic coalition game [1] with transferable utility [5]. We define the game as a pair (A, V) where A is a set of agents which formed by combination of agents with different types (*i.e* $H.Ag$, $M.Ag$, $E.Ag$, or $R.Ag$) and V is a global function (equation 1) that associates a real number to each formed coalition as the worth of that coalition.

Generally, our designed coalition game holds two main objectives. The former objective is to manage existing agents with respect to the limited number of agents in a sense that we obtain: (1) maximum level of security (minimum number of false alarm detection); (2) maximum level of coverage (optimum distribution of agents in the system); and (3) minimum time of reaching the equilibrium point (the stage in the game that agents prefer to stay in their coalition instead of changing or doing any other collaboration). Doing so, this analysis is concentrating to find the best dynamic coalition in the system considering number of involved agents and the process time. In the later objective, we concentrate on measuring the agents level of contribution in the coalition.

Considering $\mathcal{C}$ as the set of coalitions, $C \in \mathcal{C}$ would be a coalition consisting of a number of agents of different types. Let T be the set of time units. At each time unit in the system, based on the load of intruders, which is reported by the detector agent $D.Ag$, the best coalition should be formed with respect to the aforementioned two objectives of the game. Therefore, we define a global characteristic function V, expressed in equation 1, that assigns a value to each coalition (worth of coalition):

$$\begin{aligned} &V : 2^A \times T \rightarrow \mathbb{R} \\ &V(C_i, t) = \textstyle\sum_{j=1}^{n} \frac{1}{Tr-t} N_{ij}^2 + \frac{1}{t} \frac{W_{ij}}{N_{ij}+\varepsilon} \end{aligned} \quad (1)$$

$$V(C_i \cup C_j, t) \geq V(C_i, t) + V(C_j, t) \quad i \neq j \quad (2)$$

This function identifies the worth of coalition C_i at time t. The value Tr is generally the amount of time that involved agents needing to perform actions (objectively respecting a proper coalition). This value is set by the system administrator and basically reflects the level of security in order to form the best possible coalition. In general, if Tr is set to relatively small value, then the formation of the best coalition could be failed, which would lead to lower security level. On the other hand, by setting larger value for Tr, the best coalition could be formed, which overall increases the security level. The value n is the number of types of agents (here 4). N_{ij} concerns the number of involved agents of agent type j in coalition C_i and W_{ij} is the weight (agent's computational ability) of agents type j regardless of their community cardinality. By referring to this definition, we observe that at the initial moments of reporting intrusion by $D.Ag$ in the system the characteristic function provides a good incentive for agents in the sense that they form better coalitions and thus collaborate. This natural desire of agents to collaborate for obtaining more benefit remains up to the point that: (1) no agent in the environment can get more benefits by changing his current strategies (considering the other agents strategies); and (2) the characteristic function does not increase by changing the set of agents in the coalition, which is called the equilibrium point of the system. Moreover, the characteristic function V must satisfy the *superadditional characteristic* [6] expressed in equation 2. Based on our defined characteristic function we can put the game in the category of *supperadditive* games [5]. The characteristic function V respectively violates the *monotonicity characteristic* [2], [6] because different agent types have different capability levels. Most important issue in coalition games is to divide the coalition gained utility fairly among agents in the coalition. To this end, we used *Shapely Value* [9], expressing agents marginal contribution in coalitions, and decide for their following collaborative status. Excluding the fact that the function V assigns a value to each coalition, each agent has to have enough motivation to stay in the coalition, so called *Coalition Stability*.

Gradually, the involving agents, considering their gained utility, approach to a stable coalition (so called *Winning Coalition*). Once the winning coalition is formed, a subsequent shapely value (utility) would be dedicated to each contributing agent, which is equal to his expected gained utility all over the coalitions. Equations 3 and 4 are the general formulas that express the utility function, derived from [9].

$$\psi : \mathbb{N} \times \mathbb{R}^{2^{|N|}} \rightarrow \mathbb{R}^{|N|}$$

$$\phi_{Ag_i}(N, V) = \sum_{C \subset N \setminus \{Ag_i\}} P_{|C|}^{N} \times [V(C \cup \{Ag_i\}, t) - V(C, t)] \quad (3)$$

$$P_{|C|}^{N} = \frac{|C|!(N - |C| - 1)!}{N!} \quad (4)$$

$where$
$\sum_{Ag_i \in C} \phi_{Ag_i}(C, V) = V(C, t)$
$\forall\, C \subset A, \forall\, Ag_i \in C, \forall\, t > 0, \phi_{Ag_i}(C, V) > V(\{Ag_i\}, t)$

The value ϕ_{Ag_i} is the unique utility assigned to Ag_i, with respect to all possible formed coalitions. Generally the shapely value should satisfy two properties [6], defined in equation 3: (1) *Efficiency*: The utility vector exactly splits the total value; and (2) *Individual Rationality*: No agent receives less than what he could get on his own.

Expressive Example. We pick an example of our proposed agent-based model and apply our game theoretic analysis to show how accurate our proposed model works. Suppose that administrator's predefined levels of security (reflecting accuracy of false alarms detection) in the given system, which are as follows: (1) $Level\ one$: 60% accuracy ($10 \leq Tr < 20$); (2) $Level\ two$: 75% accuracy ($20 \leq Tr < 30$); and (3) $Level\ three$: 90% accuracy ($30 \leq Tr < 40$). Obviously the more the interaction period (Tr) is, the more accurate the detection result would be. In this example we have 2 detector agents ($D.Ag_1$ and $D.Ag_2$), 2 manager agents ($M.Ag_1$ and $M.Ag_2$), 4 honeyd agents ($H.Ag_1$ to $H.Ag_4$), 2 records agents ($R.Ag_1$ and $R.Ag_2$), and 2 execution agents ($E.Ag_1$ and $E.Ag_2$) with respect to the weights (computational ability) of $\{4, 5, 3, 2, 2\}$ for each type of agent. The system administrator sets Tr to 30 $msec$, which belongs to the third level of security of the system. Therefore, the objective is to check whether the system optimum equilibrium point with the best possible response time. At time space t_0, the load of intruders on $D.Ag_1$ and $D.Ag_2$ is 7 and 5, as shown in Figure 3. At time t_0, $D.Ag_1$ and $D.Ag_2$ are supported with stable coalitions C_1 and C_2. At time t_1, number of detected intruders by $D.Ag_2$ increases from 5 to 8 and consequently the coalition C_2 will not be stable any more. We investigate how honeyd agents should distribute the load among each other (see $H.Ag_4$'s behavior analysis using equations 1 and 2).

Table 1 shows V and ϕ values belong to agent $H.Ag_4$ for different steps of coalition formation. At time t_1, agent $H.Ag_1$ and $H.Ag_2$ receive higher requests from $D.Ag_1$, therefore they request more resources from the other honeyd agents that may collaborate in some other coalitions (C_1). Thus, agent $H.Ag_3$ and $H.Ag_4$ would be leaded to collaborate with them and consequently have more utility,

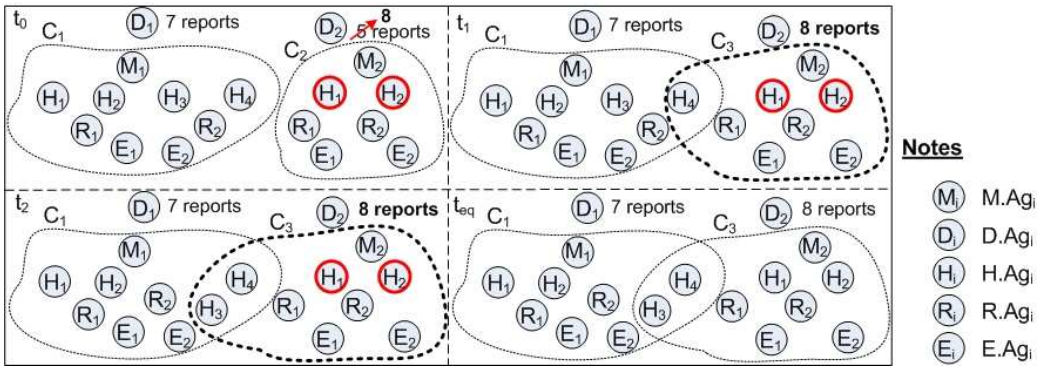

Figure 3. New coalition formation steps with new agents collaboration.

Table 2. Creation of 0.95 Confidence Intervals for the Conventional Systems and the Proposed System

Measures and Characteristics	Conventional IDS	Conventional honeyd	Proposed system
Measured ratios of false alarm	0.2265	0.1505	0.1385
	0.2345	0.146	0.131
	0.22	0.1215	0.1095
	0.195	0.1505	0.1365
	0.02135	0.1205	0.11
	0.211	0.1325	0.1205
Average ratio of false alarm	0.2167	0.1369	**0.1243**
Standard deviation of false alarm	0.013692	0.013987	0.01291
Half value of the confidence interval	0.086195	0.088052	0.081271
Full confidence interval	(0.1305,0.3029)	(0.0489,0.2250)	**(0.0431,0.01056)**

Table 1. Characteristic Function and Shapely Values over time

t_i	t_0	t_1	t_2	...	t_{eq}
$V(C_{i+1}, t_i)$	1.92	1.37	2.15	...	5.27
$\phi_4(A, V)$	0.054	0.057	0.097	...	0.19

considering the individual rationality attribute of the game. At time t_0, agent $H.Ag_4$ decides to collaborate in C_1 and generate new coalition C_3 and thus increase his utility from 0.0545 to 0.0575. Table 2 outlines the obtained values up to the new equilibrium point. In this example, system leads to the equilibrium point at time space t_{eq} which is 28 $msec$ and is less than 30 $msec$ (Tr).

4. Experimental Results

In this section, we describe the implementation of a proof of concept prototype. We also compare our proposed model with conventional IDS and Honeyd systems. In the implemented prototype, agents inherit from the basic class $Java - Simulator^{©TM}\ Agent$. The testbed environment (represented in table 2) uses legal traffic to generate different background traffic to test the system. In the simulation mostly the attack traffic is directed towards the system in an isolated test network starting at very low network load range. Therefore, we generate a high range of attacking packets, which increases the load of the system in order to evaluate the ratio of false alarm. Figure 4 compares the three systems in terms of the number of falsely detected packets, reflecting the corresponding false alarm. Figure 5 shows the results of three different tests regarding to the accuracy of the proposed system.

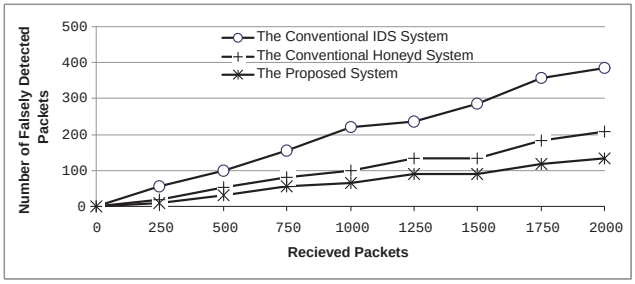

Figure 4. Performance of the three models in terms of the falsely detected packets by packet inter arrival time of 0.020.

The conventional IDS system has a high false alarm ratio because the detection accuracy is low and there is no further investigation for the suspected attacks. In the conventional

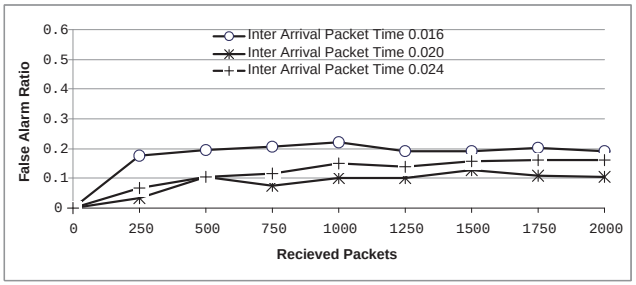

Figure 5. Performance of the proposed model in three different packet inter arrival times.

honeyd system, since there is no initial detection, it will automatically ignore the packets when the load is high, but in the proposed system since the packets are filtered initially by $D.Ag$, we do not encounter such a bottleneck at least up to very large number of attacking packets generated. Moreover, by changing the agents coalitions, the load is evenly distributed. The network load is measured as the percent of total network bandwidth occupied by the traffic. Our tests show that the proposed system can manage to improve the detection accuracy in comparison with the conventional honeyd and IDS. As it is clear, in the proposed system the objective is to re-check the attacking packets in order to avoid the false alarms while both honeyd and IDS have their own false alarm ratio, which is definitely greater than what we gained as a cooperation of these two systems. Because in the proposed system $M.Ag$ decides on the accuracy of the initial detection of $D.Ag$ and forms the best coalitions in order to investigate the suspected packets, we have a better performance of false alarm ratio in all the network loads. Table 2 represents the creation of confidence intervals for six runs. As the experimental results show, the best performance measured is 0.1243.

5. Conclusions

The paper presents a new agent-based model on a network used to decrease the false alarms rate of the intrusion detection systems. The system allocates different types of activities to each agent in the coalition based on their level of capability and their level of access to the resources (e.g. storing any necessary information about the identified attacker by records agent). We also tried to maximize level of distribution by increasing the level of autonomy and adaptivity of each type of agent. We analyzed the proposed agent-based model interactions between different agents with using cooperative game theory. Finding the best coalition of existing agents with respect to their abilities, load of attacks on the system and predefined time of response is the main purpose of our game theoretic analysis. We proposed a dynamic characteristic function in order to evaluate each coalition value considering the time of coalition formation.

For future work, we should pay more attention to the following problems: How to formalize each type of agent capabilities in order to achieve maximum level of distribution. Finding a reliable and accurate equilibrium point is one of the most challenging problems in our game theoretic analysis, which can be caused by number of dynamic impacting factors and their complicated correlation. We can extend the results of game theoretic analysis to formalize different thresholds to classify the levels of security in the system.

References

[1] R. Ariel and J. Martin. A course in game theory. MIT Press, ISBN 978-0-262-65040-3. A modern introduction at the graduate level, 1994.

[2] S. Bonnevay, N. Kabachi and M. Lamure. Agent-based simulation of coalition formation in cooperative games. Proc. of the 2005 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, pp.136-139, 2005.

[3] B. Khosravifar, and J. Bentahar. An experience improving intrusion detection systems false alarm ratio by Uusing honeypot. In Proc. of the 22nd international conference on advanced information networking and applications, pp. 997-1004, 2008.

[4] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. ACM SIGCOMM computer communications review, Vol(34)-1, 2004.

[5] K. Leyton-Brown, Y. Shoham. Essentials of Game Theory: A Concise, Multidisciplinary Introduction. Morgan and Claypool Publishers, 2008.

[6] J. Mario. Cooperative Games on Combinatorial Structures. Kluwer Academic Publishers, 2000.

[7] A. Orfila, J. Carbo and A. Ribagorda. Intrusion detection effectiveness improvement by a multiagent system. International journal of computer science and applications, Vol(2)-1, 2005.

[8] H. Otrok, M. Debbabi, Ch. Assi and P. Bhattacharya. A cooperative approach for analyzing intrusions in mobile Ad hoc networks. 27th international conference on distributed computing systems workshops (ICDCSW'07), 2007.

[9] L. S. Shapley. A value for n-person games. In Contributions to the Theory of Games, volume II, by H.W. Kuhn and A.W. Tucker, editors. Annals of Mathematical Studies v. 28, pp. 307-317. Princeton University Press.

[10] C. H. Tsang and S. Kwong. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. IEEE international conference on industrial technology, pp.51-56, 2005.

[11] W. Wang, Ch. J. Wang and Ch. Shi-fu. Dynamic hierarchical distributed intrusion detection system based on multiagent system. Proc. of the 2006 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, pp.89-93, 2006.