Multi-Agent Systems for Dynamic Forensic Investigation

Phillip Kendrick[1], Abir Jaafar Hussain[1], Natalia Criado[2]

[1]Liverpool John Moores University, Department of Computer Science
P.G.Kendrick@2012.ljmu.ac.uk, A.Hussain@ljmu.ac.uk
[2]King's College London, Department of Computer Science
Natalia.Criado@kcl.ac.uk

Abstract. In recent years Multi-Agent Systems have proven to be a useful paradigm for areas where inconsistency and uncertainty are the norm. Network security environments suffer from these problems and could benefit from a Multi-Agent model for dynamic forensic investigations. Building upon previous solutions that lack the necessary levels of scalability and autonomy, we present a decentralised model for collecting and analysing network security data to attain higher levels of accuracy and efficiency. The main contributions of the paper are: (i) a Multi-Agent model for the dynamic organisation of agents participating in forensic investigations; (ii) an agent architecture endowed with mechanisms for collecting and analysing network data; (iii) a protocol for allowing agents to coordinate and make collective decisions on the maliciousness of suspicious activity; and (iv) a simulator tool to test the proposed decentralised model, agents and communication protocol under a wide range of circumstances and scenarios.

Keywords: Forensic Investigation, Multi-Agent System, Simulator, Cyber Security

1 Introduction

Providing effective cyber security will require efficient and scalable solutions to meet the ever increasing number of needs found within modern expanded networks. Typical security solutions utilise a number of specialised systems (e.g., Intrusion Detection Systems (IDS) [1], [2], firewalls and forensic tool-kits [3]) often requiring high performance hardware to offset the cost of processing large amounts of data. These solutions often suffer from two problems that we will address in this paper: information overflow, which occurs when systems inefficiently collect all of the available information for bulk processing; and the failure to detect advanced stealthy attacks, which occurs when systems fail to observe the necessary information required for accurate attack analysis. By combining Multi-Agent Systems (MAS) [4] with intelligent information gathering techniques, improvements can be made by taking

advantage of automated forensic investigations to proactively gather the necessary data about suspicious activity[1].

A MAS, at its most basic level, can be defined as a collection of intelligent agents. Agents are independent pieces of software, often capable of working together to solve problems that could not be solved by a single agent. The following properties are characteristic of an intelligent agent [4]: (1) Autonomy - The agent's ability to act independently without any external human operator interaction. (2) Reactivity - The agent's ability to sense environmental changes and react to the situation. (3) Proactivity - The agent's ability to choose actions to achieve goals. (4) Adaptability - The agent's ability to change goals in response to unforeseen circumstances. (5) Communication & Coordination - The agent's ability to communicate with other agents to perform more complex tasks together.

IDS are deployed in an attempt to solve the problem of detecting malicious actors who perform actions without authorisation. Problems such as the costly requirements for high performance hardware [5], the inability to monitor all relevant information sources [6], the inefficiency of having to process large amounts of information flowing through a network [5], as well as structural vulnerabilities with centralised technologies make the current era of IDS ineffective.

In this paper, intrusion detection and forensic data collection techniques are combined with MAS for more effective and efficient detection in response to cyber attacks. This is performed by gathering the relevant evidence. In particular, we present a Multi-Agent model for the dynamic organisation of agents participating in forensic investigations; an agent architecture endowed with mechanisms for collecting and analysing network data; a protocol for allowing agents to coordinate and make collective decisions on the maliciousness of suspicious activity; and a simulator tool to test the proposed decentralised model, agent architecture and protocol under a wide range of circumstances and scenarios.

The remainder of this paper is organised as follows. Section 2 contains an overview of previous research in the area of Network Security and MAS. Section 3 provides details of the proposed model including a formal definition of the agent architecture and the communication protocol. Section 4 describes the simulation tool, the parameters for customising the experiments and the simulation results. Finally in Section 5 a conclusion, discussion of potential applications of the proposed model and future work is given.

2 Related Research

Shakarian et al. [7] described a cyber attribution system [8], [9] that takes into consideration different data sources and uses MAS to reason about the origin of an attack through the use of agent reasoning. The system uses information gathered about the attack as well as information gathered from a wide range of military sources to reason in-depth about the attribution of an attack. A highlighted danger of relying on external

[1] Suspicious activity is defined as any activity that does not appear to fit the norm of the network.

sources of information is the trustworthiness of the source, which must be taken into consideration. This use of external information provided an effective way to gain extra contextual information for detected attacks but was heavily reliant on previously collected and catalogued information from military sources.

Haack et al. [10] developed a hierarchical MAS model for monitoring and reporting data within the security environment. The system was composed of a number of agent types, each with a specific task to perform, such as event monitor, alert and report for system operators. The flow of information consisted of a high level policy created by a system operator which would be disseminated to the lower level agents responsible for monitoring networked components. Alerts would be generated by the monitor agents and aggregated by a higher level agent to make decisions about potential security events. The structure of this system is inherently centralised as the information, which is transferred up and down an agent pipeline, gets processed by one dedicated agent rather than having decisions made locally by the individual agents.

Jahanbin et al. [11] proposed a MAS framework for forensic information gathering which uses three types of agents for data collection, data analysis and alert generation. The authors note how the MAS paradigm is well suited to the task of forensic data collection as agents can be dispatched to areas of the network to perform collection and analysis of evidence such as log files. This system is structurally similar to Haack et al. [10] with layered agents passing information up the agent pipeline to a central agent for decision making. This central agent structure is similar to an IDS as it collects information and then makes a judgement based on that information, however, if some information is missing, the system would continue processing new information rather than actively searching missing data.

Baig et al. [12] performed a survey of the current application of MAS in a number of critical infrastructure fields including intrusion detection. Emphasis was placed on system resilience so that if the system was attacked, resulting in some agents being forced off-line, the remaining agents should reorganise themselves to continue operation. Having agents specifically designed to adapt to network changes (e.g., hosts being turned off, firewalls restricting access to a subnet or intentional compromise) was shown to be a critical consideration, especially in the security environment.

Mees [13] designed a MAS to detect Advanced Persistent Threats by using external data sources to lookup the origins of suspicious connections. Within the framework three agents were described: a consultation agent to evaluate the location of the IP address, an analysis agent to compare the suspect connection with previously seen traffic patterns and a third agent to attempt to distinguish between human and robot connections by performing task-specific analysis. By using agents in this way to gather external information from data sources located beyond the local network perimeter, the agents were able to gather extra information that might not have been available to traditional IDSs which, for security, do not usually make external connections. This system utilises agents capable of performing multiple tasks which doesn't take advantage of having a greater number of more specialised agents for improve scalability. Our model uses a greater number of specialised agents to encourage competition between agents where multiple actions could be taken at any given time.

3 Dynamic Forensic Investigation Model

To address the problems of threat detection in networking environments, we propose a Multi-Agent model that combines the traditional tasks of an IDS using forensic collection capabilities to intelligently respond to cyber attacks. By bestowing both forensic information gathering and IDS-like information analysis capabilities onto agents, the proposed model is capable of automatically following the relevant lines of digital investigation after the detection of a suspicious activity.

The current era of cyber security solutions relies on the bulk collection and processing of data to detect instances of malicious behaviour. This centralised approach often leads to an overflow of information resulting in performance bottlenecks which inhibits the accurate analysis of suspicious activity. This can be improved upon by using multiple agents for data collection and analysis, which will be located in multiple physical areas on the network, thereby reducing the amount of information flowing to one single node. The total amount of information that must be processed for a particular suspicious activity can be reduced by only actively collecting extra data when there is some evidence to suggest that the data is relevant. Instead of collecting all possible data at all times, agents should collect the minimal amount to detect specific attacks and then, if an agent suspects an attack is occurring, start to collect more data specific to that activity. This is different to the current era of IDS technologies which typically collect all available information at all times regardless of the type of suspicious activity. By performing data collection and analysis tasks using a number of agents, and only utilising their functionalities when there is cause to do so, the resulting system will be both more accurate and efficient since only necessary tasks will be performed.

Rather than having a central repository of all information collected as seen in previous systems in [10], [11], [14]–[17], in the proposed system, the information is locally stored within the individual agents. Furthermore, agents are capable of performing one data collection action using information as input when interacting with a data source[2] to obtain more information (output) about a suspicious activity. By distributing data collection tasks among the agents in this way, the processing can be kept locally near to the sources of information rather than having a potentially vulnerable central system that could be the target of attacks. This organisation in itself provides many security and performance benefits over current centralised models, for example, agents are replaceable, hence if one agent goes off-line due to an attack or poor network conditions, the remaining agents could reorganise themselves around the information gap to continue collecting data.

3.1 Model Overview

The model is formally defined as a set of agents ($G=\{g_1, ..., g_i\}$), where each agent (g_i) can perform a data collection action for gathering information about suspicious

[2] A data source is defined to be a source of information that exists, this may be an external data source such as a DNS server or a local source such as connection logs.

activities. This information can be formalised by a set of features F representing different attributes or characteristics of a given activity; e.g., the IP address of a given connection, VPN usage, etc. Each feature ($f \in F$) has a domain (D_f) containing all its possible values.

A data collection action is any forensic collection task that retrieves data from some data source; e.g., the collection of Domain Name Service (DNS) logs or a list of currently connected IP addresses. Data collection actions are represented using conditions and effects. Conditions must be satisfied to execute the action; i.e., conditions represent the information that needs to be known in order to perform a data collection action. The effects represent the new information that will be known once the data collection action has been executed successfully.

Definition 1. *A data collection action is defined as a tuple* $\langle C, e \rangle$

- C *is the action conditions; i.e., a set of pairs* (f, v) *where* $f \in F$ *and* $v \in D_f$*;*
- $e \in F$ *is the action effect; i.e., a feature whose value will be determined by the action.*

Once an agent executes its data collection action it performs an analysis of the available information for suspicious activity; e.g., analysing the geographical origin of the collected IP addresses to identify malicious connections. More formally, a data analysis action is defined as follows:

Definition 2. *Given a set of pairs (f, v) representing the available information about a suspicious activity, we define a data analysis action as a function returning a value between [0, 1] representing the probability of the suspicious activity being malicious.*

Note that the effects of a data collection action are used as conditions for other data collection actions, allowing agents to cooperate in *extended data collection process*. In particular, only those agents whose action conditions are satisfied are included in the process (i.e., those agents capable of collecting more information by using the information already collected as input). Our model includes a protocol to allow agents to coordinate when they participate in extended data collection process. The benefit of using this model is that a line of investigation only takes place when there is previous evidence (i.e., as a satisfied condition) to suggest that more useful data may be found. Traditional IDSs do not consider the context of previous information to tailor its future actions and target areas that has evidence suggesting more data might be found. The proposed model creates a chain of agent knowledge that is built up as each agent performs its own data collection which is later used to analyse the attack as a whole. Informally, we say that the conditions of a data collection action are satisfied when all the pairs feature value in the action condition are also contained by the available information.

Definition 3. *Given a set I formed by pairs (f, v) representing the available information about a suspicious activity, and a data collection action* $\langle C, e \rangle$ *we define that*

action conditions are satisfied iff for all (f, v) ∈ C, (f, v) ∈ I; and not satisfied otherwise.

At the beginning of a security event, only a small amount of information is known about the attack so agents will perform actions that will provide context to the situation. Then, agents with a small number of conditions will first perform their data collection task, simpler data collection actions will have fewer conditions on their execution. As more knowledge is acquired, more complex agents with multiple conditions will be able to perform their data collection tasks while agents that have not had their conditions satisfied will not be able to participate. This will improve the efficiency of intrusion detection by allowing agents to logically avoid certain collection tasks until such a time when there is reasonable evidence to suggest that further useful data will be found. For example, assuming that there are 100 available actions, with 50 actions that are intended to run on a remote network and 50 that are intended to run on the local network. By performing the simple action of determining whether the attacker is located either remotely or locally, the agent could reduce the number of possible actions that would need to be performed. Performing remote analysis when the attacker has been identified as being on the local network would be computationally wasteful. Using this principle we will achieve greater levels of efficiency by only collecting and analysing pieces of information that are relevant to the context of the situation.

3.2 Agent Architecture

Agents in the proposed model consist of a set of modules that provide the base functionality, each of the following modules: monitoring, intrusion flagging, data collection, local analysis, communication and global analysis module, are described in detail below.

Monitoring Module. The monitoring module takes sampling tasks (e.g., monitoring data such as incoming IP address, VPN usage, etc.), distributed across a number of agents, to monitor for different network activities. The agents can monitor a host, service or network point and the observed data will be stored locally to be used in future anomaly analysis.

Intrusion Flagging Module. The intrusion flagging module is used to analyse the monitored data obtained by the previous module against the agent's local database to make a preliminary decision on whether the data is suspicious or not. This is done through the use of either signature or anomaly detection. An *escalation threshold* ($\pi \in [0, 1]$) is used to decide whether the information deviates from the normal patterns of the system. This threshold can be set relatively low as the decision on the suspiciousness of the event in question is only used to decide whether an extended data collection process should take place. In comparison to current IDS technologies, in which the first layer of anomaly analysis is often the only layer used in the decision making process, our model allows different agents to cooperate to further investigate events.

Data Collection Module. The data collection module is used to interact with the data source that allows the agent to obtain more information about a suspicious activity. As aforementioned, a data collection action requires some known information (i.e.,

conditions) to obtain some new information (i.e., effect). Note that some data collection actions (e.g., primitive information gathering actions) may not require any conditions to produce an output.

Local Analysis Module. The output received from the data collection module will be analysed by the local analysis module, which uses the agents local database to classify the information as either malicious or not. As aforementioned, the local analysis will calculate the probability of the suspicious activity being malicious. In particular, the agent builds a *local report* containing its identity, the output of the data collection module, and its local analysis value. This local report will be used to extend the available information about the suspicious activity.

Definition 4. *A local report defined as a tuple* $\langle g, (f, v), p \rangle$ *where:*

- $g \in G$ *is the agent's identity;*
- (f, v) *is a pair feature value corresponding to the output of the data collection action performed by agent g;*
- $p \in [0, 1]$ *is the agent's analysis of the suspicious activity; i.e., the probability of the suspicious activity being malicious.*

During the local analysis, information collected from sources will be analysed using the agent's local database of results collected from random sampling and anomaly analysis. This will produce an analysis of the information, also considering external factors such as the validity of the information retrieved, the integrity of the source or the integrity of the agent itself. As an example, consider an agent that collects information from a mail server. As part of the agent's tool-kit, it has the ability to check third-party bug tracking databases to detect whether there are any known vulnerabilities in the server. Consider the case that the agent finds a critical vulnerability with the data source it is assigned to. When the agent comes to take part in an extended collection task and finds the event to be innocuous with some degree of certainty, it could choose to reduce the degree of certainty to a lower value due to the existence of a vulnerability. This is done to adapt to changes in the environment and to take into account other factors that existing systems would normally not consider.

Communications Module. The communications module is used to send and receive data from other agents to perform extended data collection processes. The communications module will send the available information to other agents to be added to the existing information by performing their own local data collection task. In particular, both the available information and the newly collected information (local report) will be combined into a *global report*.

Definition 5. *A global report R is defined as a set of local reports* $\{\langle g_1, (f_1, v_1), p_1 \rangle, \ldots, \langle g_n, (f_n, v_n), p_n \rangle\}$ *containing the information collected by different agents participating in a same extended data collection process.*

Extended data collection occurred when each agent performs its local data collection task, aggregates the newly collected data with the current global report and then sends it to the next agent.

Global Analysis Module. The global analysis module as input a global report and makes a decision about the maliciousness of the suspicious activity under investigation.

Definition 6. *Given a global report G representing the local decisions made by the agents participating in an extend data collection process, the global decision is a function returning a value between [0, 1] representing the collective judgement about the maliciousness of the investigated activity.*

Various methods could be used to aggregate the results, for example, the average or highest number of decisions could be used to arrive at different outcomes. In MAS, voting [18] is used to allow agents to make group decisions based on the individual experiences of each agent, this is used within the proposed system to aggregate the individual local decisions produced by each agent into a global decision about the maliciousness of the event. An example voting system might take the highest number of votes for either malicious or innocuous and set the global decision to that value. A threshold (*termination threshold*) is used to determine if the global decision has reached the required level of certainty to justify the end of the extended data collection process or if further investigation is needed.

3.3 Interaction Protocol

To allow agents participating in an extended data collection to coordinate our model includes an interaction protocol (depicted in Figure 1) The protocol is formed by five main phases: (i) request for participants; (ii) proposals from available participants; (iii) participant selection; (iv) inform summary; (v) inform result, described as follows.

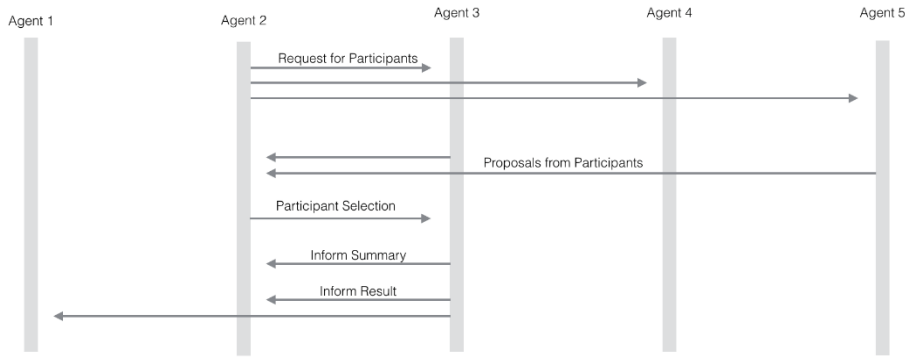

Fig. 1. An example information flow between 5 agents. Agent 2 is the current agent. Agent 2 requests participants to perform extended data collection, Agents 3 and 5 respond while Agent 4 cannot participate so ignores the request. Agent 2 selects Agent 3 to be the next agent to perform its data collection task. After the data collection task, Agent 3 sends back a summary to Agent 2, which after making the final decision, sends the result to all previous agents (Agents 1 and 2) to end the extended data collection process.

Request for Participants. Once an agent has performed its data collection and analysis tasks, the agent must add its local report to the global report and then send it onto the next agent for further information collection. The communication module is used to facilitate this.

The first step of the interaction protocol is aimed at requesting help from other agents that can participate in the data collection process. In particular, the set of pairs feature value are extracted from the global report and then broadcast (by the initiating agent) to the other agents. Given a global report $\langle g_1, (f_1, v_1), p_1 \rangle, ..., \langle g_n, (f_n, v_n), p_n \rangle$ a request is formalised as a set $\{(f_1, v_1), ...,(f_n, v_n)\}$ containing the available information about a suspicious activity.

Proposals from Participants. Any agents whose data collection action is satisfied by the information contained can respond by indicating their availability to participate in the extended data collection task.

Participant Selection. It is possible that several agents respond to the initial request indicating that they can work with the available data. The initiator must decide which agent will be selected to continue with the data collection process. In particular, the initiator will send the whole global report to this selected agent.

Unlike in other MAS solutions, the proposed model does not include a central repository of agents which can be queried to find the most suitable agent for a given task. This improves scalability but requires a system to allow agents to find each other. The agents will maintain a local database of agents that they have previously worked with. Deciding which agent should be selected as the preferred agent will affect the performance of the system as a whole. If the most optimal agent is selected for the task most of the time, the search process will improve as less time is spent performing data collection by unreliable agents. There are a number of ways in which the preferred agent can be identified based on what is important in a given situation. If accuracy is especially important for the current event the agents may select the agent that most often votes correctly, this will produce result in a more accurate search but could come at the cost of efficiency. If time is an important factor during some event the agent may choose the fastest performing agent to collect information quickly, this will produce a result faster than the previous but could potentially result in a less certain decision. While analysis of factors such as these could be done to determine the optimal preferred agent selection algorithm, events within the security environment can often be unpredictable and allowing the agents to choose the preferred agent at run time could produce a more adaptable solution.

For example, in a Distributed Denial of Service (DDoS) attack, selecting the preferred agent based on the speed at which agents are capable of collecting information could not result in a timely collection if one part of the network is under attack. How-

ever, selecting agents based on their geographical location (e.g., using the agents that are not under attack), would improve the efficiency of the response.

Inform Summary. To endow agents with information to select the preferred agent for a given task the inform summary step sends back its performance summary to the agents predecessor. This summary will be logged by the initiator agent for use in the preferred agent selection process. The parameters sent in the summary will include the agent's local decision about the maliciousness of the event, as well as, performance variables such as the time taken to perform the collection task, the importance of the data collected and the computational cost of performing the collection.

Inform Result. Once a final decision has been reached, the final decision will be sent to all of the participating agents, this can then be used by the agents to review its method for selecting the preferred agent (e.g., the preference can be increased for those agents with local decisions in-line with the final decision).

4 Simulation Tool

The proposed model has been implemented as a simulator tool for future testing. Agents are initialised with a set of binary features to simulate information about the security environment. The agents are displayed graphically as nodes and the path of the extended data collection process as it is passed between the agents as edges (see Figure 2).

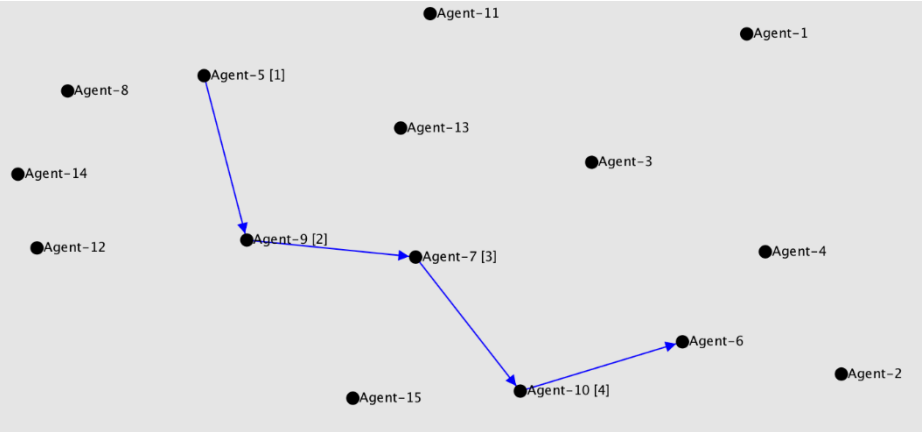

Fig. 2. An example simulation using 15 agents, 5 of which participated in the extended collection while the rest did not have their conditions satisfied so were unable to run. Nodes represent individual agents, blue arrows as well as the number in the square brackets show the order in which they ran.

The simulator consists of a number of parameters that can be configured to control all aspects of the environment and the agents. Environmental parameters such as *number of runs* and *number of iterations* control the number of times the simulation

should be repeated (reinitialising the agent settings each time) and the number of simulated events, both malicious and innocuous, that will occur per run. Agent variables controlling the number of agents, as well as the number of conditions assigned to each agent, can be configured to simulate scenarios using a small or large number of agents with a variable number of conditions that must first be satisfied before the agent can perform its collection action. The *analysis* variable is used to model the agent's ability to correctly detect whether a simulated event was malicious or innocuous. This analysis value is compared to a random assigned real number value between the [0, 1] interval to simulate noise. The higher the analysis value, the more certain an agent will be that its decision is correct. In addition, various thresholds such as for deciding when an agent should submit its local decision can be configured to set the level of certainty the agent must have before voting whether the data is malicious. Controlling this variable can be done adaptively and will be the focus of future work. Likewise the global decision threshold can be configured to test the effectiveness of different voting algorithms when calculating the agent's final global decision about the event as a whole.

Three types of log files are created during execution of the simulator, the first of which is an action level account showing the initial values for each agent as well as whether their local decision was correct or not. The second file includes an iteration level report which shows the agents involved during each simulated event, whether the global decision was correct or not and the statistics to measure the agents performance (e.g., true positive rate, false positive rate, sensitivity, specificity, etc.). The final log is saved after each run and stores averaged values for the previously mentioned agent performance statistics as well as any simulator threshold values.

5 Discussion

In this paper we have presented a multi-agent model for forensic investigation with potential applications in domains where devices are often distributed across a wide area and information is scattered between local and remote networks. Besides that, Network Security could benefit from this adaptive model as the environment is unpredictable and long term goals cannot be planned for. For example, in intrusion detection scenarios the best way to identify an attack or attacker is often heavily dependent on the attacker's actions and so cannot be defined in advance. Our model could also be used over a number of decentralised Internet of Things devices where resources are scarce. Agents could be installed onto low-end hardware since each agent only has to maintain a local database of logged data specific to their task. Finally, our model could be applied to scenarios that require adaptive information gathering or exploration tasks, as our agents agents are endowed with mechanisms to organise themselves to perform tasks without top-down or fixed instruction. Our model only requires that tasks be broken down into individual actions that can be distributed among the agents.

Future work will be focused on exploring the most optimal configurations under different scenarios and accuracy, efficiency and reliability conditions. Another direc-

tion of research involves extending the proposed model with mechanisms to take automated action in response to the detected threats.

[1] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3. pp. 26–41, 1994.

[2] T. Verwoerd and R. Hunt, "Intrusion detection techniques and approaches," *Comput. Commun.*, vol. 25, no. 15, pp. 1356–1365, 2002.

[3] M. R. Clint, M. Reith, C. Carr, and G. Gunsch, "An Examination of Digital Forensic Models," *Int. J. Digit. Evid.*, vol. 1, no. 3, pp. 1–12, 2002.

[4] M. Woolridge, *An Introduction To MultiAgent Systems*, 2nd ed. Wiley, 2011.

[5] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2012.

[6] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, J. Van Bokkelen, and S. Enterprises, "Forensics Analysis," no. December, pp. 60–66, 2002.

[7] P. Shakarian, G. I. Simari, G. Moores, and S. Parsons, "Cyber Attribution : An Argumentation-Based Approach," *arXiv Prepr.*, pp. 151–171, 2015.

[8] P. Shakarian, G. I. Simari, G. Moores, S. Parsons, M. A. Falappa, E. Engineering, C. Science, U. S. M. Academy, and W. Point, "An Argumentation-Based Framework to Address the Attribution Problem in Cyber-Warfare," *arXiv Prepr.*, 2014.

[9] G. I. S. M. Shakarian, Paulo, "Belief revision in structured argumentation," *Found. Inf. Knowl. Syst.*, pp. 324–343, 2014.

[10] J. N. Haack, G. a. Fink, W. M. Maiden, a. D. McKinnon, S. J. Templeton, and E. W. Fulp, "Ant-based cyber security," *Proc. - 2011 8th Int. Conf. Inf. Technol. New Gener. ITNG 2011*, pp. 918–926, 2010.

[11] A. Jahanbin, A. Ghafarian, S. A. H. Seno, and S. Nikookar, "A Computer Forensics Approach Based on Autonomous Intelligent Multi-Agent System," *Int. J. Database Theory Appl.*, vol. 6, no. 5, pp. 1–12, Oct. 2013.

[12] Z. a. Baig, "Multi-agent systems for protecting critical infrastructures: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 1151–1161, 2012.

[13] W. Mees, "Multi-agent anomaly-based APT detection," *Proc. Inf. Syst. Technol. Panel Symp.*, pp. 1–10, 2012.

[14] N. Afzali Seresht and R. Azmi, "MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 286–298, 2014.

[15] F. Alkhateeb, Z. A. Al-fakhry, E. Al Maghayreh, S. Aljawarneh, and A. T. Al-taani, "A Multi-Agent-Based System for Securing University," vol. 2, no. March, pp. 223–231, 2010.

[16] A. Orfila, J. Carbo, and A. Ribagorda, “Intrusion Detection Effectiveness Improvement by a Multi-agent System,” *Int. J. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 1–6, 2005.

[17] G. Helmer, J. S. K. Wong, V. Honavar, L. Miller, and Y. Wang, “Lightweight agents for intrusion detection,” *J. Syst. Softw.*, vol. 67, no. 2, pp. 109–122, 2003.

[18] S. R. and P. Norvig, *Artificial Intelligence: International Version: A Modern Approach*. 2010.