Multiagent Systems for Network Intrusion Detection: A Review

Álvaro Herrero and Emilio Corchado

Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio,escorchado}@ubu.es

Abstract. More and more, Intrusion Detection Systems (IDSs) are seen as an important component in comprehensive security solutions. Thus, IDSs are common elements in modern infrastructures to enforce network policies. So far, plenty of techniques have been applied for the detection of intrusions, which has been reported in many surveys. This work focuses the development of network-based IDSs from an architectural point of view, in which multiagent systems are applied for the development of IDSs, presenting an up-to-date revision of the state of the art.

Keywords: Multiagent Systems, Distributed Artificial Intelligence, Computer Network Security, Intrusion Detection.

1 Introduction

Firewalls are the most widely used tools for securing networks, but Intrusion Detection Systems (IDSs) are becoming more and more popular [1]. IDSs monitor the activity of the network with the purpose of identifying intrusive events and can take actions to abort these risky events. A wide range of techniques have been used to build IDSs, most of the reported and described in previous surveys [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15].

On a more general context, the actual demands of effectiveness and complexity have caused the development of new computing paradigms. Agents and multiagent systems (MAS) [16] are one of these new paradigms. The concept of agent was originally conceived in the field of Artificial Intelligence (AI), evolving subsequently as a computational entity in the software engineering field. From the software standpoint, it is seen as an evolution to overcome the limitations inherent to the object oriented methodologies. Up to now, there is not a strict definition of agent [17]. In a general AI context, a rational agent was defined [18] as anything that perceives its environment through sensors and acts upon that environment through effectors. In a more specific way, a software agent has been defined as a system with capacity of adaptation and provided with mechanisms allowing it to decide what to do (according to their objectives) [19]. Additionally, from a distributed AI standpoint, it was defined [20] as a physical or virtual entity with some features: capable of acting in an

Á. Herrero et al. (Eds.): Computational Intel. in Security for Info. Systems, AISC 63, pp. 143–154.
springerlink.com

environment, able of communicating directly with other agents, possessing resources of its own, and some others.

It is in a multiagent system (MAS) that contains an environment, objects and agents (the agents being the only ones to act), relations between all the entities, a set of operations that can be performed by the entities and the changes of the universe in time and due to these actions" [20]. From the standpoint of distributed problem solving [21] a MAS can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver. According to [22], the characteristics of MASs are:

- Each agent has incomplete information, or capabilities for solving the problem, thus each agent has a limited viewpoint.
- There is no global system control.
- Data is decentralized.
- Computation is asynchronous.

As a consequence of that, agents in a MAS are driven by their own objectives as there is not a global control unit. They take the initiative according to their objectives and dynamically decide what to do or what tasks other agents must do.

Agents and multiagent systems have been widely used in last years, not always being the most appropriate solution. According to [23], there is a number of features of a problem which point to the appropriateness of an agent-based solution:

- The environment is open, or at least highly dynamic, uncertain, or complex.
- Agents are a natural metaphor. Many environments are naturally modelled as societies of agents, either cooperating with each other to solve complex problems, or else competing with one-another.
- Distribution of data, control or expertise. It means that a centralised solution is at best extremely difficult or at worst impossible.
- Legacy systems. That is, software technologically obsolete but functionally essential to an organisation. Such software cannot generally be discarded (because of the short-term cost of rewriting) and it is often required to interact with other software components. One solution to this problem is to wrap the legacy components, providing them with an "agent layer" functionality.

Since its inception in the 1980s, IDSs have evolved from monolithic batch-oriented systems to distributed real-time networks of components [10]. As a result, new paradigms have been designed to support such tools. Agents and multiagent systems are one of the paradigms that best fit this setting as ID in distributed networks is a problem that matches the above requirements for an agent-based solution. Furthermore, some other AI techniques can me combined with this paradigm to make more intelligent agents.

This paper surveys and chronologically analyses previous work on multiagent systems for network intrusion detection (Section 2), emphasizing the mobile agent approach (Section 3).

2 IDSs Based on Agents

One of the initial studies under this frame was JAM (Java Agents for Metalerning) [24]. This work combines intelligent agents and data mining techniques. When applied to the ID problem, an association-rules algorithm determines the relationships between the different fields in audit trails, while a meta-learning classifier learns the signatures of attacks. Features of these two data mining techniques are extracted and used to compute models of intrusion behaviour.

In the 90's DARPA defined the Common Intrusion Detection Framework (CIDF) as a general framework for IDS development. The Open Infrastructure [25] comprises a general infrastructure for agent based ID that is CIDF compliant. This infrastructure defines a layered agent hierarchy, consisting of the following agent types: Decision-Response agents (responsible for responding to intrusions), Reconnaissance agents (gather information), Analysis agents (analyse the gathered information), Directory/Key Management agents, and Storage agents. The two later provide support functions to the other agents.

AAFID (Autonomous Agents For Intrusion Detection) [26] is a distributed IDS architecture employing autonomous agents, being those defined as "*software agents that perform a certain security monitoring function at a host*". This architecture defines the following main components:

- **Agents:** monitor certain aspects of hosts and report them to the appropriate transceiver.
- **Filters:** intended to be the data selection and abstraction layer for agents.
- **Transceivers:** external communications interfaces of hosts.
- **Monitors:** the highest-level entities that control entities that are running in several different hosts.
- **User interface:** interact with a monitor to request information and to provide instructions.

This architecture does not detail the inner structure or mechanisms of the proposed agents, that use filters to obtain data in a system-independent manner. That is, agents do not depend on the operating system of the hosts. Additionally, AAFID agents do not have the authority to directly generate an alarm and do not communicate directly with each other.

In [27], a general MAS framework for ID is also proposed. Authors suggest the development of four main modules, namely the sniffing module (to be implemented as a simple reflex agent), the analysis module (to be implemented as several agents that keeps track of the environment to look at past packets), the decision module (to be implemented as goal-based agents to make the appropriate decisions), and the reporting module (to be implemented as two simple reflex agents: logging and alert generator agents). These components are developed as agents:

- The sniffing agent sends the previously stored data to the analysis agents when the latter request new data. One analyser agent is created for each one of the attacks to be identified. They analyse the traffic reported from the sniffing module, searching for signatures of attacks and consequently building a list of suspicious packets.

- Decision agents are attack dependant. They calculate the severity of the attack they are in charge from the list of suspicious packets built by analyser agents. Decision agents also take the necessary action according to the level of severity.
- Finally, the logging agent keeps track of the logging file, accounting for the list of suspect packets generated from the decision agents. On the other hand, the alert generator agent sends alerts to the system administrator according to the list of decisions.

Some topics about ID based on MASs are briefly discussed in [28], where a general framework for ID is proposed. Such framework includes the following classes of agents: learning data management agents, classifier testing agents, meta-data forming agents, and learning agents.

SPIDeR-MAN (Synergistic and Perceptual Intrusion Detection with Reinforcement in a Multi-Agent Neural Network) is proposed in [29]. Each agent uses a SOM and ordinary rule-based classifiers to detect intrusive activities. A blackboard mechanism is used for the aggregation of results generated from such agents (i.e. a group decision). Reinforcement learning is carried out with the reinforcement signal that is generated within the blackboard and distributed over all agents which are involved in the group decision making.

An heterogeneous alert correlation approach to ID by means of a MAS is proposed in [30]. In this study alert correlation refers to the management of alerts generated by a set of classifiers, each of them trained for detecting attacks of a particular class (DoS, Probe, U2R, etc.). Although it is a Host-based IDS (HIDS), the main idea underlying the design of this MAS could be also applied to Network-based IDSs (NIDSs). According to the adopted Gaia methodology, roles and protocols are specified in this study. The roles are mapped into the following agent classes:

- **NetLevelAgent (DataSensor role):** in charge of raw data preprocessing and extracting both events and secondary features.
- **BaseClassifiers (DecisionProvider role):** performs source based classification and produces decisions after receiving events from sources. Several subclasses are defined to cover the different predefined types of attacks and the different data sources.
- **Metaclassifiers (DecisionReceiver and DecisionProvider roles):** one agent of this class is instantiated for each one of the attack types. They combine decisions produced by the BaseClassifiers agents of the assigned attack type.
- **SystemMontor (ObjectMonitor role):** visualises the information about security status.

CIDS (Cougaar-based IDS) [31] provides a hierarchical security agent framework, where security nodes are defined as consisting of four different agents (manager agent, monitor agent, decision agent, and action agent) developed over the Cougaar framework [32]. It uses intelligent decision support modules to detect some anomalies and intrusions from user to packet level. The output of CIDS (generated by the Action Agent) consists on the environment status report (IDMEF format [33]) as well as recommendations of actions to be taken against the ongoing intrusive activities. The system employs a knowledgebase of known attacks and a fuzzy inference engine to classify network activities as legitimate or malicious.

PAID (Probabilistic Agent-Based IDS) [34] is a cooperative agent architecture where autonomous agents perform specific ID tasks (e.g., identifying IP-spoofing attacks). It uses three types of agents:

- **System monitoring agents:** responsible for collecting, transforming, and distributing intrusion specific data upon request and evoking information collecting procedures
- **Intrusion-monitoring agents:** encapsulate a Bayesian Network and performs belief update using both facts (observed values) and beliefs (derived values). They generate probability distributions (beliefs) over intrusion variables that may be shared with other agents, which constitutes the main novelty of PAID. Methods for modelling errors and resolving conflicts among beliefs are also defined.
- **Registry agents:** coordinate system-monitoring and intrusion-monitoring agents.

A multiagent IDS framework for decentralised intrusion prevention and detection is proposed in [35]. The MAS structure is tree-hierarchical and consists of the following agents:

- **Monitor agents:** capture traffic, preprocess it (reducing irrelevant and noisy data), and extract the latent independent features by applying feature selection methods.
- **Decision agents:** perform unsupervised anomaly learning and classification. To do so, an ant colony clustering model is deployed in these agents. When attacks are detected, they send simple notification messages to corresponding action and coordination agents.
- **Action agents:** perform passive or reactive responses to different attacks.
- **Coordination agents:** aggregate and analyse high-level detection results to enhance the predictability and efficiency.
- **User Interface agents:** interact with the users and interpret the intrusion information and alarms.
- **Registration agents:** allocate and look up all the other agents.

A MAS comprising intelligent agents is proposed in [36] for detecting probes. These intelligent agents were encapsulated with different AI paradigms: support vector machines, multi-variate adaptive regression, and linear genetic programming. Thanks to this agent-based approach, specific agents can be designed and implemented in a distributed fashion taking into account prior knowledge of the device and user profiles of the network.

By adding new agents, this system can be easily adapted to an increased problem size. Due to the interaction of different agents, failure of one agent may not degrade the overall detection performance of the network.

MOVIH-IDS (Mobile-Visualization Hybrid IDS) [37] is built by means of a MAS that incorporates an artificial neural network for the visualisation of network traffic. It includes deliberative agents characterized by the use of an unsupervised connectionist model to identify intrusions in computer networks. These deliberative agents are defined as CBR-BDI agents [38], [39] using the Case-based Reasoning paradigm [40] as a reasoning mechanism, which allows them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and to have a large capacity for adaptation to the needs of their surroundings.

3 Mobile Agents

Apart from the above works, some others have focused on the mobile-agent approach. That is, agents travelling along different hosts in the network to be monitored. Some issues about the application of mobile agents to ID are further discussed in [41], and examples following this approach are described in this section.

IDA (ID Agent) system [42] is aimed at detecting many intrusions efficiently rather than accurately detecting all intrusions. To do so, it approaches ID from a novel standpoint: instead of continuously monitoring the activity of users, it watches events that may relate to intrusions (MLSI – Mark Left by Suspected Intruders). When an MLSI is detected, IDA collects further information, analyses it and decides whether an intrusion has taken place. To do so, two kinds of mobile agents contribute to the information collection stage: a tracing agent is sent to the host where suspicious activity comes from and once there, it activates an information-gathering agent. Several information-gathering agents may be activated by several different tracing agents on the same target system.

Micael [43] was proposed as an IDS architecture based on mobile agents. Its main difference to previous proposals is the task division. ID tasks are distributed to the following agent kinds: head quarter (centralizes the system's control functions), sentinels (collect relevant information, and inform the head quarter agents about eventual anomalies), detachments (implement the counter-measures of the IDS), auditors (check the integrity of the active agents), and special agents with different duties. By moving throughout the network, the mobile auditor agents can audit each of the defended hosts sequentially.

Mobile agents are applied in [44] to make critical IDS components resistant to flooding DoS and penetration attacks. To do so, the attacked IDS component will be automatically relocated to a different (still operational) host. This relocation is invisible to the attacker who then cannot persist in the attack. Every critical agent has one or more backup agents (maintaining full or partial state information of the agent they are backing up) that reside on distinct hosts within the same domain. When the machine hosting a critical agent is down (whatever the reason is), its backup agents contact each other to decide on a successor that will resume the functions of the original agent. One of the main drawbacks of this solution is that temporarily the network may be partially unprotected while the IDS critical components are moving from one host to another.

SPARTA (Security Policy Adaptation Reinforced Through Agents) is proposed in [45] as an architecture to collect and relate distributed ID data using mobile agents. According to the authors, SPARTA mobile agents enable the distributed analysis, improve the scalability, and increase the fault tolerance. Some security issues about these mobile agents are considered. The required information (interesting host events) for event correlation is locally collected and stored, which is considered a distributed database with horizontal fragmentation. Mobile agents are in charge of collecting the distributed information (matching a given pattern) to answer user queries.

SANTA (Security Agents for Network Traffic Analysis) [46] is proposed as a distributed architecture for network security using packet, process, system, and user information. It attempts to emulate mechanisms of the natural immune system using IBM's Aglets agents. The proposed monitoring agents roam around the machines (hosts or routers) and monitor the situation in the network (i.e., look for changes such as malfunctions, faults, abnormalities, misuse, deviations, intrusions, etc.). These immunity-based agents can mutually recognize each other's activities and implement Adaptive Resonance Theory neural networks and a fuzzy controller for ID. According to the underlying security policies and the information from the monitoring agents, decision/action agents make decisions as to whether an action should be taken by killer agents. These killer agents terminate processes that are responsible for intrusive behaviour on the network.

A distributed ID architecture, completed with a data warehouse and mobile and stationary agents is proposed in [47]. The MAS is combined with a rule generation algorithm, genetic algorithms, and datawarehouse techniques to facilitate building, monitoring, and analysing global, spatio-temporal views of intrusions on large distributed systems. System calls executed by privileged processes are classified after being represented as feature vectors. To do so, different agents are defined:

- **Data cleaner agents:** these stationary agents process data obtained from log files, network protocol monitors, and system activity monitors into homogeneous formats.
- **Low-level agents:** these mobile agents form the first level of ID. They travel to each of their associated data cleaner agents, gather recent information, and classify the data to determine whether suspicious activity is occurring. These agents collaborate to set their suspicion level to determine cooperatively whether a suspicious action is more interesting in the presence of other suspicious activity.
- **High-level agents:** they maintain the data warehouse by combining knowledge and data from the low-level agents. The high-level agents apply data mining algorithms to discover associations and patterns.
- **Interface agent:** it directs the operation of the agents in the system, maintains the status reported by the mobile agents, and provides access to the data warehouse features.

In [48] a multiagent IDS (MAIDS) architecture containing mobile agents is proposed. These lightweight agents, located in the middle of the architecture, form the first line of ID. They periodically travel between monitored systems, obtain the gleaned information, and classify the data to determine whether singular intrusions have occurred.

In the MA-IDS architecture [49] mobile agents are employed to coordinately process information from each monitored host. Only the critical components in the MA-IDS architecture (Assistant and Response agents) are designed as mobile agents. An Assistant mobile agent is dispatched by the Manager component to patrol (gather information) in the network. Assistant mobile agents are intended to determine whether some suspicious activities in different hosts are part of a distributed intrusion. If that is the case, the Manager component will possibly dispatch a Response mobile agent to "intelligently" response to each monitored host. It is claimed that these mobile agents are capable of evading attackers and resurrecting themselves when attacked. Additionally, agent mobility makes distributed ID possible by means of data correlation and cooperative detection.

An interesting and comprehensive discussion about optimising the analysis of NIDSs through mobile agents is presented in [50]. The main proposal is to place the mobile analyser components of the NIDS closer together in the network and shifting the processing load to underused nodes if possible.

APHIDS (Agent-Based Programmable Hybrid Intrusion Detection System) [51] implements the distributed search and analysis tasks with mobile agents equipped with scripting capability to automate evidence gathering. This architecture is similar to the SPARTA and MAIDS systems (all of them exploit the mobility of the agents to perform distributed correlation), but APHIDS allows the specification of coordinated analysis tasks using a high-level specification language. Mobile agents are used for monitoring the output from other previously running IDSs (HIDSs or NIDSs), querying the log files and system state, and reporting results.

APHIDS was subsequently upgraded, generating APHIDS++ [52] that introduces a two-level caching scheme:

- Task Agents enter the first level cache mode (busy wait at the attacked machine) after having handled an initial attack. Each Task Agent maintains a publicly accessible queue of pending attacks to handle.
- If no new attacks are sent to a Task Agent within a certain time limit, the agent enters the second cache level mode, in which it is flushed to its host machine's disk. Thus, resource consumption in the host machine is reduced.

Some other improvements of APHIDS++ are the addition of an optional intelligent agent and an XML implementation of the Distributed Correlation Script.

Two different agent classes are proposed in [53]: monitoring agents (AM) and managing agents (AZ). AM observe the nodes, process captured information, and draw conclusions for the evaluation of the current state of system security. AM agents can travel along the network to monitor different areas that may be at risk of attacks. On the other hand, AZ agents are responsible for creating profiles of attacks, managing AM agents, and updating its database and ontology.

IDReAM (Intrusion Detection and Response executed with Agent Mobility) is proposed in [54] as a new approach to build a completely distributed and decentralized ID and Response System in computer networks. Conceptually, IDReAM combines mobile agents with self-organizing paradigms inspired by natural life systems: immune system that protects the human body against external aggressions and the stigmergic paradigm of a colony of ants. The two natural systems exhibit a social life by the organisation of their entities (immune cells and ants) which the author states is not possible without mobility. IDReAM is assessed in terms of resource consumption and intrusion response efficiency.

IDSUDA (Intrusion Detection System Using Distributed Agents) [55] proposes the application of mobile agents to monitor the usage of various system resources in order to detect deviations from normal usage. The behaviour of the attacker is tracked by following up the intruder movements from one resource to another.

A general distributed IDS framework based on mobile agents is proposed in [56]. Some of the components in such model are designed as mobile agents for the purpose of high adaptability and security of the system. It is claimed that these mobile agents can evade intrusion and recover by themselves if they suffer from intrusion, but further explanations of that are not provided.

4 Conclusions

A great effort has been devoted to the MAS approach for Intrusion Detection. Plenty of works have been released on this subject, enabling the ID task in complex and distributed environments. The use of dynamic MASs enables taking advantage of some of the properties of agents such as reactivity, proactivity, and sociability. One of the main weaknesses of such solutions is the defence mechanisms of the MASs, as the resistance to attacks has not been considered in most previous work.

Although mobile agents can provide an IDS with some advantages (mobility, overcoming network latency, robustness, and fault tolerance), some problems have not been completely overcome yet [41]: speed, volume of the code required to implement a mobile agent, deployment, limited methodologies and tools, security threats, and so on.

Acknowledgments. This research has been partially supported by the projects BU006A08 of the JCyL and CIT-020000-2008-2 of Spanish Ministry of Science and Innovation. The authors would also like to thank the manufacturer of components for vehicle interiors, Grupo Antolin Ingeniería, S.A. in the framework of the project MAGNO 2008 - 1028.- CENIT Project funded by the Spanish Ministry of Science and Innovation.

References

1. Chuvakin, A.: Monitoring IDS. Information Security Journal: A Global Perspective 12(6), 12–16 (2004)
2. Frank, J.: Artificial Intelligence and Intrusion Detection: Current and Future Directions. In: 17th National Computer Security Conf., Baltimore, MD, vol. 10 (1994)
3. Mukherjee, B., Heberlein, L.T., Levitt, K.N.: Network Intrusion Detection. IEEE Network 8(3), 26–41 (1994)
4. Engelhardt, D.: Directions for Intrusion Detection and Response: a Survey. Electronics and Surveillance Research Laboratory, Defence Science and Technology Organisation, Department of Defence, Australian Government (1997)
5. Jones, A., Sielken, R.: Computer System Intrusion Detection: A Survey. White paper. University of Virginia - Computer Science Department (1999)
6. Debar, H., Dacier, M., Wespi, A.: Towards a Taxonomy of Intrusion-Detection Systems. Computer Networks - the International Journal of Computer and Telecommunications Networking 31(8), 805–822 (1999)
7. Axelsson, S.: Intrusion Detection Systems: A Survey and Taxonomy. Technical Report. Chalmers University of Technology. Department of Computer Engineering (2000)
8. Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., Stoner, E.: State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99-TR-028. Carnegie Mellon University - Software Engineering Institute (2000)
9. McHugh, J.: Intrusion and Intrusion Detection. International Journal of Information Security 1(1), 14–35 (2001)
10. Verwoerd, T., Hunt, R.: Intrusion Detection Techniques and Approaches. Computer Communications 25(15), 1356–1365 (2002)

11. Mukkamala, S., Sung, A.H.: A Comparative Study of Techniques for Intrusion Detection. In: 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 570–577 (2003)
12. Estevez-Tapiador, J.M., Garcia-Teodoro, P., Diaz-Verdejo, J.E.: Anomaly Detection Methods in Wired Networks: a Survey and Taxonomy. Computer Communications 27(16), 1569–1584 (2004)
13. Lazarevic, A., Kumar, V., Srivastava, J.: Intrusion Detection: a Survey. In: Managing Cyber Threats: Issues, Approaches, and Challenges 5. Massive Computing, pp. 19–78. Springer, US (2005)
14. Patcha, A., Park, J.-M.: An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Computer Networks 51(12), 3448–3470 (2007)
15. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges. Computers & Security 28(1-2), 18–28 (2009)
16. Wooldridge, M., Jennings, N.R.: Agent theories, architectures, and languages: A survey. Intelligent Agents (1995)
17. Franklin, S., Graesser, A.: Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents. In: Jennings, N.R., Wooldridge, M.J., Müller, J.P. (eds.) ECAI-WS 1996 and ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997)
18. Russell, S.J., Norvig, P.: Artificial Intelligence: a Modern Approach. Prentice Hall, Englewood Cliffs (1995)
19. Weiss, G.: Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (1999)
20. Ferber, J.: Multi-agent Systems: an Introduction to Distributed Artificial Intelligence. Addison-Wesley, Reading (1999)
21. Durfee, E.H., Lesser, V.R.: Negotiating Task Decomposition and Allocation Using Partial Global Planning. In: Distributed Artificial Intelligence, vol. 2. Morgan Kaufmann Publishers Inc., San Francisco (1989)
22. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1(1), 7–38 (1998)
23. Wooldridge, M.: Agent-based Computing. Interoperable Communication Networks 1(1), 71–97 (1998)
24. Stolfo, S., Prodromidis, A.L., Tselepis, S., Lee, W., Fan, D.W., Chan, P.K.: JAM: Java Agents for Meta-Learning over Distributed Databases. In: Third International Conference on Knowledge Discovery and Data Mining, pp. 74–81 (1997)
25. Reilly, M., Stillman, M.: Open Infrastructure for Scalable Intrusion Detection. In: IEEE Information Technology Conference, pp. 129–133 (1998)
26. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. Computer Networks: The International Journal of Computer and Telecommunications Networking 34(4), 547–570 (2000)
27. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. IEEE Potentials 22(4), 28–31 (2003)
28. Gorodetski, V., Kotenko, I., Karsaev, O.: Multi-Agent Technologies for Computer Network Security: Attack Simulation, Intrusion Detection and Intrusion Detection Learning. Computer Systems Science and Engineering 18(4), 191–200 (2003)
29. Miller, P., Inoue, A.: Collaborative Intrusion Detection System. In: 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS 2003), pp. 519–524 (2003)

30. Gorodetsky, V., Karsaev, O., Samoilov, V., Ulanov, A.: Asynchronous alert correlation in multi-agent intrusion detection systems. In: Gorodetsky, V., Kotenko, I., Skormin, V.A. (eds.) MMM-ACNS 2005. LNCS, vol. 3685, pp. 366–379. Springer, Heidelberg (2005)
31. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsettii, R.: CIDS: An agent-based intrusion detection system. Computers & Security 24(5), 387–398 (2005)
32. Cougaar: Cognitive Agent Architecture, `http://cougaar.org/`
33. Debar, H., Curry, D., Feinstein, B.: The Intrusion Detection Message Exchange Format (IDMEF). IETF RFC 4765 (2007)
34. Gowadia, V., Farkas, C., Valtorta, M.: PAID: A Probabilistic Agent-Based Intrusion Detection system. Computers & Security 24(7), 529–545 (2005)
35. Tsang, C.-H., Kwong, S.: Multi-agent Intrusion Detection System in Industrial Network using Ant Colony Clustering Approach and Unsupervised Feature Extraction. In: IEEE International Conference on Industrial Technology (ICIT 2005), pp. 51–56 (2005)
36. Mukkamala, S., Sung, A.H., Abraham, A.: Hybrid Multi-agent Framework for Detection of Stealthy Probes. Applied Soft Computing 7(3), 631–641 (2007)
37. Herrero, Á., Corchado, E., Pellicer, M.A., Abraham, A.: MOVIH-IDS: A Mobile-Visualization Hybrid Intrusion Detection System. Neurocomputing 72(13-15), 2775–2784 (2009)
38. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-Based Reasoning Technology. International Journal of Intelligent Systems 18(12), 1227–1241 (2003)
39. Pellicer, M.A., Corchado, J.M.: Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1), 25–32 (2005)
40. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39–59 (1994)
41. Jansen, W.A., Karygiannis, T., Marks, D.G.: Applying Mobile Agents to Intrusion Detection and Response. US Department of Commerce, Technology Administration, National Institute of Standards and Technology (1999)
42. Asaka, M., Taguchi, A., Goto, S.: The Implementation of IDA: An Intrusion Detection Agent System. In: 11th Annual Computer Security Incident Handling Conference, vol. 6 (1999)
43. De Queiroz, J.D., da Costa Carmo, L.F.R., Pirmez, L.: Micael: An Autonomous Mobile Agent System to Protect New Generation Networked Applications. In: Second International Workshop on Recent Advances in Intrusion Detection, RAID 1999 (1999)
44. Mell, P., Marks, D., McLarnon, M.: A Denial-of-service Resistant Intrusion Detection Architecture. Computer Networks: The International Journal of Computer and Telecommunications Networking 34(4), 641–658 (2000)
45. Krügel, C., Toth, T., Kirda, E.: SPARTA: a Mobile Agent Based Instrusion Detection System. In: IFIP TC11 WG11.4 First Annual Working Conference on Network Security: Advances in Network and Distributed Systems Security. IFIP Conference Proceedings, vol. 206, pp. 187–200. Kluwer, Dordrecht (2001)
46. Dasgupta, D., Brian, H.: Mobile Security Agents for Network Traffic Analysis. In: DARPA Information Survivability Conference & Exposition II (DISCEX 2001), vol. 2, pp. 332–340 (2001)
47. Helmer, G., Wong, J.S.K., Honavar, V.G., Miller, L.: Automated Discovery of Concise Predictive Rules for Intrusion Detection. Journal of Systems and Software 60(3), 165–175 (2002)
48. Helmer, G., Wong, J.S.K., Honavar, V., Miller, L., Wang, Y.: Lightweight Agents for Intrusion Detection. Journal of Systems and Software 67(2), 109–122 (2003)

49. Li, C., Song, Q., Zhang, C.: MA-IDS Architecture for Distributed Intrusion Detection using Mobile Agents. In: 2nd International Conference on Information Technology for Application (ICITA 2004), pp. 451–455 (2004)
50. Marks, D.G., Mell, P., Stinson, M.: Optimizing the Scalability of Network Intrusion Detection Systems Using Mobile Agents. Journal of Network and Systems Management 12(1), 95–110 (2004)
51. Deeter, K., Singh, K., Wilson, S., Filipozzi, L., Vuong, S.T.: APHIDS: A mobile agent-based programmable hybrid intrusion detection system. In: Karmouch, A., Korba, L., Madeira, E.R.M. (eds.) MATA 2004. LNCS, vol. 3284, pp. 244–253. Springer, Heidelberg (2004)
52. Alam, M.S., Gupta, A., Wires, J., Vuong, S.T.: APHIDS++: Evolution of A programmable hybrid intrusion detection system. In: Magedanz, T., Karmouch, A., Pierre, S., Venieris, I.S. (eds.) MATA 2005. LNCS, vol. 3744, pp. 22–31. Springer, Heidelberg (2005)
53. Kolaczek, G., Pieczynska-Kuchtiak, A., Juszczyszyn, K., Grzech, A., Katarzyniak, R.P., Nguyen, N.T.: A mobile agent approach to intrusion detection in network systems. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 514–519. Springer, Heidelberg (2005)
54. Foukia, N.: IDReAM: Intrusion Detection and Response Executed with Agent Mobility Architecture and Implementation. In: Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005). ACM, The Netherlands (2005)
55. Alim, A.S.A., Ismail, A.S., Ahmed, S.H.: IDSUDA: An Intrusion Detection System Using Distributed Agents. Journal of Computer Networks and Internet Research 5(1), 1–11 (2005)
56. Wang, H.Q., Wang, Z.Q., Zhao, Q., Wang, G.F., Zheng, R.J., Liu, D.X.: Mobile agents for network intrusion resistance. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) APWeb Workshops 2006. LNCS, vol. 3842, pp. 965–970. Springer, Heidelberg (2006)