# Integrated Multi-Agent Approach to Network Security Assurance: Models of Agents' Community

V. GORODETSKI[1], I. KOTENKO[1], and V. SKORMIN[2]
*[1] Russia, St.-Petersburg Institute for Informatics and Automation*
*Email: gor@mail.iias.spb.su, ivkote@robotek.ru*
*[2] USA, Binghamton University*
*Email: vskormin@binghamton.edu*

Key words: Network security, security modeling, multi-agent systems, intrusion detection, access control, authentication, machine learning.

Abstract: In this paper, an integrated multi-agent approach to construction of Network Security System (NSS) is considered. The NSS consists of a multitude of specialized intelligent agents that are distributed over the computer network. The architecture of the NSS is outlined. Emphasis is given to a description of the operation and learning mechanisms implemented in the security agents.

## 1. INTRODUCTION

*Network Security assurance* is a highly complex problem due to the growing size and interconnectivity of networks, high number of users, increasing number of vulnerable targets, and the appearance of new types of effective attacks.

Presently, to provide integrated network security the sets of security subsystems are used. These subsystems are components of the operating system, applications and independently operating specialized security systems. This approach, however, becomes inefficient for modern networks. It requires formation of gigantic databases and consumes excessive amount of resources, causing inflexibility of software and hardware facilities of the network. It does not provide the required protection from the attacks and cannot adapt to new types of attacks.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*

According to a modern vision of the network security problem, particular protective components must be integrated within a global system, and distributed between the hosts of the network. The specialized components (agents) of such an *network security system (NSS)* must interact via message exchange in order to make decisions. In addition, protective components must be able to learn unknown types of intrusions and to adapt to the network reconfiguration and to traffic variations.

In the paper, an integrated multi-agent NSS is described. The paper is organized as follows. *Section 2* presents the justification of a multi-agent approach to the NSS design. In *Section 3*, an outline of the architecture of a multi-agent NSS is given. *Sections 4 and 5* are devoted to the description of basic security agents. The principles of agents' interaction and cooperation are presented in *Section 6*. *Section 7* describes the learning mechanisms of NSS. *Section 8* is devoted to the description of the NSS modeling and evaluation. *Section 9* presents a short overview of the existing research, outlines the results and directions of the future efforts.

## 2. MULTI-AGENT TECHNOLOGY FOR SECURITY

One of the most promising approaches to the NSS implementation is the utilization of *intelligent multi-agent technology*. The latter is justified by the following reasons. (1) *Uncertainty of the scripts of an attack and distribution in time its phases*. Many of attacks are realized following pre-set scripts, which consist of the consequent phases aiming to overcome protections at different network levels. By parsing jointly, the suspicious operations can be unambiguously interpreted. For this reason the interaction of various specialized security programs is necessary. (2) *Sophisticated attacks on networks have a distributed character* in the sense that they can be carried out at once through several hosts of the network and have the intrusion goal of penetrating a set of hosts. For this reason the interaction of security systems located on various hosts of networks is necessary. (3) *Security objects and security tools in networks are distributed* in confines of a host, and in confines of a network. It dictates the necessity to have a distributed federated security system. (4) *Each security subsystem has the specific rather autonomously realizable functions*, co-operating with the security objects, "supported" behind it, and with other security subsystems. For this reason it is necessary to have autonomous security subsystems capable to interact. (5) *Interaction of security subsystems can be carried out at various levels of solutions*. Such style of interaction determines a fruitful way of decomposition of security functions and a necessity of interaction between various security subsystems. It is naturally modeled in the terms of multi-agent system. (6) The built-in system of learning new attacks and their detection is necessary. In view of the large variety of security objects and types of attacks *the system of learning should consist of a set of specialized learning programs*, which also should cooperate among themselves. (7) *Security subsystems should be mobile,* for example, in the case of multistage authentication procedures fulfilled at information exchange between various network hosts.

## 3. ARCHITECTURE OF AGENTS' COMMUNITY

An NSS is viewed as a cooperative community of agents, distributed in the network. It includes agents of the following types [4]: (1) *Intrusion detection agents (IDA)* performing their tasks on the basis of available information about normal functioning processes, typical user profiles, possible anomalies, unauthorized access channels and probable scripts of attacks. (2) *Access control agents (ACA)* providing access to the information and software resources to the appropriate users. (3) *Identification and authentication agents (IAA)* responsible for the identification of sources of information and confirmation of their authenticity. These agents maintain the conformity between the implemented functional processes and the subjects initiated by these processes. (4) *Attack suppression agents (ASA)* responsible for "prosecution" and neutralization of the intruders. (5) *Damage control agents (DCA)* responsible for assessing the inflicted damage and information recovery. (6) *Cryptography and steganography agents (CSA)* responsible for the safety of the communication channels between the network nodes (hosts, servers). (7) *Learning agents (LA)* responsible for the adaptation (learning) of particular NSS agents to network reconfiguration and new types of attacks. (8) *Meta-agents (MA)* managing information security processes and policies, that results in the coordinated and cooperated behavior of the individual agents and assures the required level of general security according to a global criteria or policies. (9) *Agents-daemons (D)* intended for monitoring and preprocessing traffic data.

## 4. INTRUSION DETECTION AGENTS

Intrusion Detection Subsystems (IDS) of NSS are distributed on hosts of the network. Each host-based IDS comprises a number of the specialized IDA and the meta-agent that, acting in co-operation, perform the intrusion detection task. Each IDA performs the intrusion detection task within its host, and participates in attack detection on the entire network within its area of competency. An IDA makes decisions on the basis of data received from agent-daemons associated with respective entry points, the host-based meta-agent, ACA and IAA. As soon as an IDA detects a suspicious behavior or intrusion at its respective entry point, it forwards an appropriate message to the host-based meta-agent. The area of the IDA competence is formed by knowledge base. It contains models of attacks and information facilitating the operation within the area of the IDA responsibility. An important component of IDA is an inference engine that performs the classification of the input connections of the host and/or system calls. Inference engines utilize binary decision trees in the following fashion.

Consider an IDA having index *i* (*i=1,2,...,I*), responsible for the detection of attack types that constitute the multitude *(i)*. Assume *(i) = {a1,..., an, u}*, where *a1,...,an* are the identifiers of "known" attacks and symbol *u* corresponds to an *"unknown"* attack. Let us map *(i)* on the root node of the decision tree of the IDA. Each node of the IDA decision tree is mapped by a subset of *(i,k)* *(i)*, where *k* is the index of the decision tree node. The IDA decision tree constitutes a partially ordered family subset of the set of attacks *(i)* corresponding to the IDA area of responsibility.

The decision-making procedure corresponds to a step-by-step ascending from one node of the decision tree to another to a leaf. Each step of this procedure corresponds to an alternative choice of the subset of attacks. The choice is made through the inference in the local knowledge base attached to the current node of the decision tree. Hence, the knowledge base of each IDA is a tree-ordered set of local knowledge bases. Output information of each IDA is forwarded to the host-based meta-agent responsible for the final decision that determines the consequent behavior and communication with other components of the NSS.

The entire multitude of the IDA knowledge bases forms a distributed knowledge base of the Intrusion Detection System (IDS) of a host. It is assumed that within the described NSS architecture, an IDS is installed in a fashion that involves an agent-based learning system to assist in the development of local knowledge bases of an IDS.

## 5. ACCESS CONTROL, IDENTIFICATION AND AUTHENTICATION AGENTS

The *ACA* carry out two basic functions: (1) operate the flow of information with various degrees of confidentiality, not allowing the outflow of sensitive information on the non-authorized access channels; (2) provide access to information resources in strict conformity with set policies. The first function is implemented by means of performance of mandatory access control rules (ACR), and the second one – by discretionary ACR. The access control subsystems of NSS consist of the mandatory and discretionary access control subsystems, including the mandatory and discretionary ACR check agents, and the mandatory and discretionary ACR consistency support agents. The *mandatory ACR check agents* manage significant events (such as initialization, completion of processes, message transfer) on conformity with mandatory ACR. The check is based on the internal representation of the information processes as the *trusted and active process identifier bases*. The *discretionary ACR check agents* inspect correspondence of the messages from subjects to objects on the basis of *discretionary ACR base.* The base represents multilevel data structure. The rules of the concrete depth of an access control correspond to each level of this structure. The *mandatory and discretionary ACR consistency support agents* inspect mismatches of the discretionary and mandatory ACR during input of discretionary ACR by the security manager.

The *IAA* set correspondence between the functional components of the entire network (its subjects and objects) and the actual processes of information transmission and processing. For authentication, the services, submitted by the cryptography protection agents are used: calculation of hash function and implementation of the asymmetric cryptography channel. *The asymmetric cryptography channel* assumes that the message source has a couple of keys: the *private key* known only for a source and the *public key* known for all receivers. An public key can decrypt the message enciphered by a private key.

The *identification and authentication subsystem* consists of a set of the IAA. The *agents of loading* permit initialization of the functional processes from the pre-set software components. The *certificate creation agents* form the certificate for each process using the *base of certificates*. The *process certification agents* write the process certificate in the *active process certificates base*. At each call of the process to other process, the certification agent determines the responsible principal's names, using the certificate of the process, and transmits these names to discretionary ACA. From pair of keys of asymmetric encoding, the public key represents the principals in an authentication subsystem. The *base of keys and principal names* is used for verification of correspondence of a key and logical name of the principal. The authentication agents cooperate by the exchange of process certificates. If the process transfers the message to the process on the other host, then the authentication agent of the other host inquires about the message certificate for the source verification. For messaging between processes of various hosts, the *channel control agents* create the "process-process" channels by multiplexing the cryptography channel between hosts.

## 6. AGENTS' COOPERATION

When necessary, the behavior of host-based agents is coordinated by a host-based meta-agent. The community of host-based meta-agents, learning agents and the network-based meta-agent achieves interaction and cooperation of agents within the entire network. All interactions are performed in the message exchange mode. Output messages of the host-based meta-agent are self-initiated or constitute responses to input messages.

Message exchange surmises that agents are able in some sense to "understand" each other. Mutual agent understanding implies that each agent: (1) "knows" the agent that should be addressed to request help, if source agent's functionality and/or information are not sufficient for dealing with a problem within its scope of responsibility, (2) "knows" the agent(s) to whom the result of the performed task must be directed, and (3) agent's messages must be represented in a form and in terms that are understood to the addressees. These aspects are summarized as the presence of a *common context* or *divided knowledge* in NSS.

One of the most promising approaches to establishing a model of the distributed agents' knowledge, beliefs and common ground of the entire multi-agent system is the utilization of domain *ontology* [6]. The ontology characterizes the knowledge domain, disregarding particular structures of knowledge representation, inference algorithms or heuristics. The ontology of the network security domain is a description of the partially ordered concepts of this domain and the relationships over these concepts, which should be used by the appropriate security agents [4].

The specialization of an agent is reflected by a subset of nodes of the ontology that represents the knowledge of the respective agent. Some nodes of the ontology can be common for more than one agent. Many times only one of these agents has the detailed description of the node. This agent is the "owner" of the appropriate component of knowledge base. At the same time, there is a part of an ontological knowledge base that is common for all agents, including the meta-agent. This part of knowledge plays the role of a common context (knowledge) of the NSS.

## 7. MECHANISMS OF LEARNING

Agent-based Learning System assures the learning ability and adaptability of an NSS. The learning ability is the ability of an NSS to use its experience for detecting an intrusion at an entry point of a network. The adaptability of an NSS is its ability to detect attacks of unknown types and to learn to identify and distinguish them from other types of unknown attacks.

Data Mining and Knowledge Discovery from Data (KDD) techniques provide a fruitful basis for solving the intrusion detection tasks [1,3,5,9,10,11]. Existing research, experience and numerical experiments indicate that intrusion detection learning is supported by the classes of algorithms listed below [10]: (1) *Classification*: it converts each record of the audit data into a category of a predefined class ("normal", "abnormal", "category of attacks", "unknown", etc.). These algorithms define a classifier in the form of *rules* and *decision tree*, possibly, with an assigned measure of uncertainty. (2) *Link analysis:* it aims at finding the relationships between fields in the audit database records. These relationships form a basis allowing for the specification of user profiles in terms of *association rules*. (3) *Sequence analysis:* it is used to develop a model of time-based sequential event patterns that frequently occur together.

The *general scheme of the commonly accepted learning technology* is relatively standard. *First*, the link and sequence analyses are used for the extraction of useful patterns from pre-processed audit data. *Then*, on the basis of these patterns a rule-based classification algorithm is formulated. Because of the specifics of audit data, the first step does not have a standard implementation technique and all known approaches need to be considered. In contrast, the second step could be solved by implementing a standard tool (for example, by the RIPPER algorithm [3]).

Implementation of any learning technology for intrusion detection is a very complex task. Indeed, audit data gathered from a server during just one day may contain more than one million records of different length. Consequently, providing both accuracy and appropriate computational complexity of the learning procedure and the resultant classifier constitutes a formidable task. To provide the required accuracy and efficiency, authors of the JAM Project [10] proposed a so-called "meta-classification" approach. Briefly, the concept of this approach is as follows. The learning system is equipped with a number of simple classification algorithms ("base classifiers") capable of learning using its own training and testing data. They may be very simple and computationally efficient but, as a rule, not very accurate. After learning, these classifiers are tested on a new audit data, and each testing record is submitted to all base classifiers. While performing classification of the same interpreted audit record, each base classifier makes its own decision about category of input record. Decisions of all base classifiers are incorporated in a new record that is supplemented by the correct interpretation. These records form data for training a meta-classifier.

There are many advantages of intrusion detection utilizing meta-classification. The approach results in much more efficient and accurate classifiers. It allows for adding new base classifiers, learning techniques and audit learning and testing data to update the IDS. Meta-classifiers may be structured hierarchically thus allowing for constructing multi-level IDA consistent with the proposed IDS architecture.

The most outstanding advantage of the meta-classification approach is that it is naturally suitable for learning to detect network attacks. Indeed, each IDS can use its own audit data associated with different entry points of the network. For example, one IDS receives and processes *IP*-packets audit data to detect attack on network protocol, another IDS analyzes the *login* and *password* to detect eventual attack on the host, another IDS performs the processing of the server input (WWW, FTP, etc.). A meta-classifier combines evidences of a number of host-based IDS and forms the global policy of a network defense system.

According to the agent-based architecture proposed, the learning task is assigned to the Learning System that is responsible for IDA learning independently at every host in which they are based. The upper layer of the architecture coincides with the meta-agent of the Learning System. It manages learning within the entire network via communication between the host-based agents and the network-based agents. It exchanges messages with the above meta-agents and makes decisions on the necessity of updating the knowledge base of an IDA, on generating a new IDA, on cloning a new host-based IDS, etc. To perform the above tasks, it manages the operation of Learning Agents that are responsible for the installation of IDA, updating their knowledge bases, etc. The meta-agent of the Learning System manages the multitude of base classifiers and meta-classifiers implemented as specialized agents interacting within the Learning System. The success of the development of the entire NSS depends critically on the efficiency and accuracy of the Learning System.

## 8. NSS MODELING AND EVALUATION

To demonstrate the validity of our approach we are in the process of developing a *modeling testbed of an NSS* and its evaluation. The hardware of the testbed includes a local computer network that has access to Internet. The software is based on Unix and Windows NT and a specialized agent-based program package that is being developed in Visual C++. For the first step of the testbed realization, the attack intrusion detection environment was built. It includes facilities for network attack modeling, agents-daemons, IDA, ACA, IIA and meta-agents.

The *expert evaluation of metrics of an NSS* is carried out. As main property of NSS we assume a guaranteed level of security assessed by probability $P_{NAA}$ of non-authorized access (NAA). The dependency obtained makes it possible to define the main tendency to increase the access control depth by ACA under decreasing of quality of operation of other security agents. The IDA determines the probability $P_{chan}$ of the intrusion channel existence depending on the power of rules of the IDA knowledge base. During the beginning phase of the IDS operation its knowledge base may be poor. Later on, if a quantity of rules is accumulated through learning procedure $P_{chan}$ approximately can be assessed as inversely proportional to the amount of rules. The increase of the access control depth causes a growth of time needed to execute the mandatory ACR. The time of the ACR execution is a part of total duration of the functional operations. Instead, the intrusion detection related operations could be executed both in real time and in the delayed time compared to the protected functional processes. Therefore, to be compared with access control system, the IDS slows the protected functional processes to a lesser degree. This means that agent-based approach makes it possible a more flexible management by access control and intrusion detection functions.

## 9. RELATED WORKS AND CONCLUSION

Recently there were a number of publications and projects exploiting the idea of distributed multi-agent NSS, see [7, 8, 12, 13]. There exist a few papers, for example, [2, 13] that consider an agent-based approach for an NSS design, however only intended primarily for the intrusion detection task. All referred papers have some common limitations: (1) they consider only one information security related task, the intrusion detection, and ignore all other tasks; (2) they do not pay the needed attention to the agent cooperation and multi-agent architecture.

In our research, these shortcomings are the primary focus of attention. The main features of the proposed approach are as follows: (1) Agent-based NSS architecture which is extendable, adaptive and may be implemented efficiently; (2) Approach to agents cooperation and message interpretation that is based on the use of ontology of the information security domain which is considered as the framework for distributed common knowledge; (3) Agents' learning subsystem architecture that is implemented by using a set of "base classifiers" and a meta-classifier. Each base classifier is used as a node of a decision tree having its own knowledge base and responsible for the detection of a subset of attacks including *"unknown"* types of attacks.

Future plans include the development of the "learning by feedback" methods in more detail. Furthermore, affirm such NSS properties as real-time extensibility and adaptability. The latter would allow an NSS to withstand new kinds of network attacks and variability of network structure and platforms.

## REFERENCES

1. R.Agrawal, T.Imielinski, A.Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993.
2. J.Balasubramaniyan, J.Garcia-Fernandez, D.Isakoff, E.Spafford, D.Zamboni. An Architecture for Intrusion Detection using Autonomous Agents. In *Proceedings of the 14th Annual Computer Security Applications Conference*. Phoenix, Arizona. 1998.
3. W.W.Cohen. Fast effective rule induction. In *Machine Learning: the 12th International Conference*, Lake Taho, CA, 1995.
4. V.I.Gorodetski, L.J.Popyack, I.V.Kotenko, V.A.Skormin. Ontology-based Multi-agent Model of Information Security System. In *Lecture Notes in Artificial Intelligence*, vol.1711. 1999.
5. V.Gorodetski, O.Karsaev. Algorithm of Rule Extraction from Learning Data. In *Proceedings of the 8th International Conference "Expert Systems Application & Artificial Intelligence"" (EXPERSYS-96).* IITT International, Paris, France. 1996.
6. N.Guarino. Formal ontology, conceptual analysis and knowledge representation. In *Int. J. Human-Computer Studies*, No.43, 1995.
7. Hochberg et al. "NADIR": An Automated System for Detecting Network Intrusion and Misuse. In *Computers and Security*, vol.12, No.3, 1993.
8. S.Kumar, E.H.Spafford. A software architecture to support misuse intrusion detection. In *Proceedings of the 18th National Information Security Conference*, 1995.
9. T.Lane, C.E.Brodley. Sequence matching and learning in anomaly detection for computer security. In *Proceedings of the AAAI Workshop: AI Approaches to Fraud Detection and Risk Management*. AAAI Press, 1997.
10. W.Lee, S.J.Stolfo, K.Mok. A Data mining Framework for Building Intrusion Detection Model. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1999.
11. H.Mannila, H.Toivonen. Discovering generalised episodes using minimal occurrences. In *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, Portland, Oregon, August 1996.
12. V.Paxon. Bro: A system for detecting network intruders in real time. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.
13. G.White, E.Fish, U.Pooch. Cooperating Security Managers: A Peer-Based Intrusion Detection System. In *IEEE Network*, January/February 1996.