Task #2

2.1. Title of the Project / Number of Annual Report

Project #1994P: "Formal Methods for Information Protection Technology"

Task #2: "Mathematical Foundations, Architecture and Principles of Implementation of Multi-Agent Learning Components for Attack Detection in Computer Networks".

Annual Report #2

2.2. Contracting Institute

St. Petersburg for Informatics and Automation of the Russian Academy of Sciences

2.3. Participating Institutes

None

2.4. Project Manager, phone number, fax number, e-mail address

Dr. Oleg V. Karsaev, tel:+7 (812)-323-3570, fax:+7(812)-328-0685, e-mail: ok@iias.spb.su

2.5. Commencement Date, Duration

December 1, 2000, 36 months

2.6. Brief description of the work plan: objective, expected results, technical approach

Brief description of the work plan

B-1. Development of the learning task ontology, allocation of learning tasks over generic learning agents	1-3 Quarters
Interim Report #1 summarizing the results of the task B-1	3 Quarter
B-2. Development of architecture of the Multi-agent Learning System and mathematical methods realizing learning functionalities of the generic agents	4-6 Quarters
Submission a paper in an International Journal	5 Quarter
Interim Report #2 summarizing the results of the task B-2	6 Quarter
B-3. Development of the protocols of inter-level intelligent agent interaction (negotiation), generalization of the particular agent decisions according to the meta-classification approach and development of architecture of the Multi-agent Learning System as a whole	7-8 Quarters
B-4. Development of object-oriented conceptual project of the Multi-agent Learning System	6-8 Quarters
B-5. Development of the software prototype of the Multi-agent Learning System implementing theoretical results of research and its evaluation	9-11 Quarter
Interim Report #3 describing the results of the task B-4 and partially developed software prototype of the components of the Multi-agent Learning System	10 Quarter
B-6. Evaluation of the properties, advantages and disadvantages of the developed architecture and mathematical methods implemented within the prototype of the intelligent Multi-agent Learning system	12 Quarter
Final report, summarizing the results of simulation of the developed Multi-agent Learning System aimed at attack detection, as well as final conclusion concerning the	12 Quarter

research results on the task 2 as a whole	

Notice: The grey shaded rows correspond to the tasks to be solved during the second year research.

Objective

The objectives of Task #2 of this project is a development of mathematical foundation, agent-based architecture, and principles of implementation of the Intrusion Detection Learning Systems operating in a parallel with the Computer Network Assurance System.

Expected results

1. Learning task ontology.
2. Allocation of learning tasks over generic learning agents and development of the architecture of their interaction within Multi-agent Learning System.
3. Mathematical basis and algorithms realizing learning functionalities of the particular agents and software prototypes of the components of the Multi-agent Learning System based on theoretical results of the research.
4. Simulation-based evaluation of the properties, advantages and disadvantages of the developed multi-agent model and architecture of the Multi-agent Learning system aimed to support adaptability and learnability of the Network Security System.

Technical approach

A key issue of an intrusion detection learning task is a selection and development (if necessary) of accurate and efficient methods and algorithms specialized for the data structures specifying intrusions. This data specifying is represented mainly in the form of timely ordered sequences of audit data of various lengths specified in terms of repeatable symbols. These symbols correspond to the preprocessed messages of the input traffic at a host port of the computer network. In case of the distributed attack as well as in case of the normal distributed users' activity, a set of such sequences specifies a user activity. Mining of such data is a challenge in knowledge discovery from databases.

The approaches to be used within the last task are threefold. The first class of them is based on the specification of each input sequence (example) corresponding to a class of attacks or normal activity as a word of an unknown formal Context Free language. In this case the learning task may be reduced to a task of the inductive formal grammar recovery. The second class of approaches to be used is based on statistical properties of the examples specifying normal and abnormal activity of the user(s) accessing to network resources. The third class of methods intends to solve the task of the rule extraction from the training data sample and its result is specified in terms of predicates given over higher level concepts like patterns.

An important issue is a decomposition of the whole intrusion detection learning task into multitude of sub-tasks according to the ontology of attacks, and their allocation to specialized learning software agents. Intrusion detection learning system will be implemented on the basis of multi-agent architecture. Within it, each specialized learning agent is considered as the generic one which is capable to extract patterns of a particular class (association rules, frequent patterns, production rules given over patterns, etc.), and it should be able to extract knowledge from data of the fixed format (event sequences, sets of patterns, subset of rules, etc.).

For the purpose of the formal specification of learning agent interactions, which are especially necessary in learning detection of distributed attacks, the idea of meta-classification will be used. This idea entails the necessity of distributed learning which corresponds to the multi-level learning aiming at fusion knowledge resulted from local learning procedures realized by particular learning agents.

2.7. Technical progress during the first year (for 2nd annual reports)

Technical progress during the first year was fully compliant with regard both to the tasks predefined by the Work plan and to the schedule of their completion.

Achievements of the first year

The basic achievements of the first year corresponded to the tasks scheduled. These tasks and respective results are described below.

Within the task B-1 "Development of the learning task ontology, allocation of learning tasks over generic learning agents" the following subtasks were solved:

1. Analysis of the structure of the training data.

2. Development of the multi-level ontology of the learning tasks.

3. Development of conceptual model of the generic agents of the multi-agent learning system.

Also the part of the task B-2 "Development of Architecture of the Multi-agent Learning System and Mathematical Methods Realizing Learning Functionalities of the Generic Agents" scheduled for 4–6 Quarters was solved partially.

Thus, in more detail, the main results obtained during first year research are as follows.

1. Analysis of the structure and other peculiarities of the training data.

Conceptual analysis of any learning problem includes, first of all, analysis of the sources of the knowledge. Within the task in question the main sources of knowledge are historical interpreted audit data containing sequences of user' activities ("examples") that can correspond to "*normal*", "*abnormal*" or "*interpreted abnormal*" data. (In the last case it is supposed that the class of attack is determined definitely). In the Project, intrusion detection system is considered as a multi-sensor knowledge-based data fusion system. As a rule, information from a single source does not contain much evidence that allows of timely and efficient detection of attacks and security policies violations. In order to construct an efficient intrusion detection system and a learning system, it is necessary to utilize an interconnected complex of audit data received from multiple sources and representing data from different levels of generalization (on the network level, OS level, application level, and additional sources level). Addressing multiple information sources may significantly increase the validity of decisions related to attack detection and network security. The respective learning task is considered as distributed multi-level data mining and knowledge discovery problem to be implemented based on multi-agent architecture.

The known attack detection learning mechanisms are computationally complex. The complexity can be significantly reduced through the preliminary analysis and identification of the most representative and informative attributes of the subjects' (including malefactors') activities (both internal and external to the system under protection), that are registered in the audit data. Among such attributes are repeated instances and combinations (patterns) of events; mistyped commands; indications of exploitation of the known vulnerabilities, illegal parameters, irregularities in the network traffic parameters and contents; substantial discrepancies in the values of attributes that characterize the system subjects' operations profile (e.g., time and date of activity, subject's address, services used by subjects, system resources characteristics including CPU usage of data, RAM/HDD/files/ports access data, etc.), and unexplained problems (e.g., router failure, server overload, failure to launch a system service, etc.). Involvement of experts at this stage of learning could substantially cut down the pattern search and dimensions of data needed for learning.

The Main peculiarities of the problem in question are conditioned by the temporal and distributed nature of the processes to be learned and data reflecting these processes. In the Project, this nature of the training and testing data is specified in terms of unified model of temporal sequences of events and patterns to be mined from the audit data.

2. Development of the multi-level ontology of the learning tasks

Consistent operation of a large scale distributed system, which makes decisions in a knowledge-based fashion, could only be provided if distributed entities were capable to "understand" each other. The efficient way to achieve mutual understanding of distributed entities (in our case – agents) is to use *ontology-based approach* representing the shared knowledge of distributed entities. The ontology forms the necessary basis for local knowledge bases consistency, distributed knowledge base integrity and correct interpretation of the messages entities exchange with.

The multi-level ontology of the intrusion detection learning problem unites a structured multitude of basic notions. This ontology encompasses the notions from several domain ontologies, i.e. from the "Data fusion" & "Data fusion learning" problem domain ontology, from the "Intrusion detection" and "Intrusion Detection Learning" subject domain ontologies. The ontology developed serves as the basis in the design and implementation of the upper-level representation of distributed knowledge base. This level of knowledge provides, on the one hand, for the integrity of the distributed knowledge base, and on the other hand, for the "mutual understanding" of the agents interacting via messages exchange.

3. Development of conceptual model of the generic agents of the multi-agent learning system

The learned intrusion detection system is viewed as a multi-sensor multi-level data fusion system. This system makes decisions on the basis of a multi-level model of input data (network input traffic and/or audit data) processing. Based on this viewpoint, the conceptual model of the multi-agent learning system has been developed.

On the basis of analysis of the learning data structure a number of adequate learning methods (algorithms) that according to the authors' opinion should be efficient in solving the learning tasks are selected. The chosen multitude of methods includes both some of the widely known methods (e.g., from the multitude of them including *ID3*, *C4.5*, *AQ*, *CN2*, boosting, and the meta-classification methodology) and also methods that have been or are being developed by the authors (e.g., *Visual Analytical Method –VAM*, *GK2* algorithm, *Algebraic Bayes Networks*).

The above mathematical fundamentals allowed to develop a conceptual model of the intrusion detection learning system. In the research a multi-agent architecture of intrusion detection learning system is decided. Generic agent classes of the multi-agent intrusion detection learning system under development have been determined, as well as their functions and roles in the learning system. The set of agent classes includes class of learning data management agents; class of classifier testing agents; class of meta-data forming agents; and class of learning agents.

2.8. Technical progress during the year of reference

Technical progress during the year of reference is fully compliant with regard both to the tasks predefined by the Work plan and to the schedule of their completion.

Achievements of the year of reference

The basic achievements of the year of reference correspond to the tasks scheduled. These tasks are as follows:

1. Development of architecture of the Multi-agent Learning System and mathematical methods realizing learning functionalities of the generic agents.

2. Development of the protocols of inter-level intelligent agent interaction (negotiation), generalization of the particular agent decisions according to the meta-classification approach and development of architecture of the Multi-agent Learning System as a whole.

3. Development of object-oriented conceptual project of the Multi-agent Learning System.

The main results obtained during the second year research are as follows.

1. Analysis and development of the formal model and architecture of the particular generic agents of the multi-agent learning system.

The following two tasks were the subjects of activity.

The first task concerns with the development of the mathematical and algorithmic basis supporting intrusion detection learning. Within this task the main efforts were focused on the study of the existing methods and algorithms which have been published recently and currently are used by the specialists in the area of interest. Also, the subjects of study were other methods and algorithms developed within knowledge discovery from databases area, which can potentially be used for intrusion detection learning. In this study, the main attention was paid to the methods of frequent pattern mining, in particular, mining frequent patterns from temporal sequences. At the same time, several methods were implemented with the purpose of preliminary evaluation of their properties and usefulness in the learning task of interest. In particular, such kind of the software is developed for *FP-growth* algorithm aiming at mining frequent patterns and association rules from transactional databases. Also the adaptation of the methods and algorithms developed by the authors of this research and analysis of their capabilities within intrusion detection learning task was carried out. In particular, experimental software for *GK2* method for mining rules from relational databases was developed. This software together with the Visual Analytical Mining (*VAM*) method also developed by the authors was tested on the basis of the intrusion detection training data that were used in KDD Cup-99 competition of data mining and knowledge discovery programs. The exploration of these methods is carried out within meta-classification learning task that aims at learning fusion of the local decisions produced on the basis of local sources of data resulting from monitoring security aspects of a computer network.

The second task aims at the development of formal models and architectures of the particular classes of agent forming multi-agent learning system under development. Such models and architectures were developed for the following classes of agents:

- Class of learning data management agents;
- Class of classifier testing agents;
- Class of meta-data forming agents; and
- Class of learning agents which, in turn, comprises the following sub-classes:
  i) Class of *agents* designed for the learning classifiers of attacks, whose input data are represented as temporally ordered sequences of events, and
  ii) Class of *agents* designed for the learning of classifiers that extract knowledge from learning data represented in the form of attribute vector.

Each of these classes comprises reusable and particular components. The standard components are responsible for receiving syntactic analysis and sending the messages to get exchanged between the agents, and also the standard parts of the semantic mechanisms of the input and output messages processing that is implemented based on *state machine* framework (according to the *UML* language terminology). "Individuality" of each agent class is reflected by specifications of the particular state machines (alphabets of inputs, states and transitions, state transition functions and corresponding state machine actions specified in terms of the behavior scenarios and something else), and also by concrete contents of data- and knowledge bases. The formal models of all standard components of agent classes are developed. Some of them are tested as software components. Also some models of the special components of the agent formal models and architectures are developed. All the above models are specified formally in terms of *USE CASE DIAGRAM*, which corresponds to a visual representation of the components functional behavior specified formally in terms of the *UML* formal language.

2. Analysis of computer network intrusion detection learning (IDL) task and determination of specific problems of IDL.

Intrusion detection learning (IDL) task differs from typical data mining and knowledge discovery one in many respects. The main peculiarities leading to several specific problems of IDL technology result from the peculiarities of learning data. Analysis proved that these peculiarities result from *distributed* nature and *heterogeneity* of data. The traces of illegitimate activity of users (erroneous commands of users and attacks against computer and its information resources) are reflected in multiple distributed and heterogeneous data sources (*tcpdump*, OS system calls audit trail, application audit data, etc.). The data can be represented in different data structures (relational, sequential, temporal sequential) and measured in different measurement scales (Boolean, categorical, linear ordered, real), be of different accuracy and reliability, they may be incomplete and uncertain, contain missing values, etc. These properties along with other ones put specific problems within IDL system design and implementation.

The most important and difficult problems are as follows: (1) the necessity to provide monosemantic understanding of the terminology used by different components of IDL system, (2) entity identification problem, (3) problem of diversity of data measurement scales of training data components. These problems and some other form together a so-called "data non-congruency problem". The problems are analyzed, and approaches capable to cope with these problems within IDL task are proposed.

The basis for solving the *first problem* is "ontologism", i.e. ontology-centered approach. The technology for the development and implementation of application ontology is proposed. This technology supports the development of shared (common for all software components) and private (held only by particular agents of IDL system) components of application ontology that are "coherent" with the intrusion detection problem ontology. In addition, the ontology-centered approach makes it possible to resolve several other problems.

The *second problem* corresponds to a so-called *entity identification problem*. Each local data source specifies an entity (object to be classified) only partially. Its complete specification is made up of data fragments distributed over the data sources. Therefore, a mechanism to identify such fragments is needed to make it possible to retrieve, collect and analyze together distributed data about the same user activity. It is worth noticing that some fragments of data associated with the above entity can be absent in some sources. Within this project, the *entity identification problem* is solved in the following manner. In the application ontology, for each entity, the notion of entity identifier ("*ID entity*") is introduced. This entity identifier is a primary key of that entity (in analogy with the primary key of a table). For each above identifier, a rule within the framework of the application ontology is defined, which can be used to calculate the key value. For example, a unique combination of a set of this entity attributes can be one of such rules. A rule is defined at the level of each local data source, and this rule must uniquely connect the entity identifier and the local primary key in this source. This rule specifies:

- how to derive the local primary key (from the value of which the values of all the entity attributes in this local source can be further derived) from the entity identifier value;
- how to derive the entity identifier value from the value of the local primary key of the source.

Next, the data specifying the same entity can be represented in different sources in terms of different data structures (images, signals, expert statements assigned a measure of uncertainty, scalar data presented in Boolean, categorical, ordered and/or numerical measurement scales, etc.).

And *the last* but not the least problem is that the sets of attributes of different data sources may be overlapping. At that the *same* or "*similar*" properties of the entity can be presented in different data sources in different ways, for example, in different measurement scales, with different accuracy, and so on. This problem is solved by the appropriate strategy of fusion of data of different sources that is known as fusion decisions produced on the basis of local data sources.

The respective approaches are developed with regard to all above-mentioned problems. In the multi-agent architecture of IDL system the "data non-congruency problem" is solved due to the use of special agents that are *Data source managing agents* (associated with the particular data sources) and *KDD master agent*, which is a component of meta-level part of multi-agent system. The idea of using of such kinds of agents in IDL system architecture is new and seems promising.

3. Analysis of training and testing data for intrusion detection learning.

The analysis of intrusion detection learning task permitted to determine how distributed and heterogeneous data must be processed jointly, and what requirements must be met by methods of data mining and KDD from architectural point of view. The mathematical methods of data mining and KDD themselves that cover the needs of IDL task can further be determined on the basis of analysis of structures of training and testing data peculiar for available data sources. The results of this analysis are shortly presented below.

The training and testing data for the IDL system exists in many forms and can be received from different data sources. *The taxonomies of these data sources* can be formed by different tags: (1) location of source and software generating data; (2) processing level; and (3) an object, the data are associated with:

- *The taxonomy, which classifies data sources due to location of source and software generating data*, includes two main sources: network-based, and host-based. The network-based sources depend on network layers and used protocols. Host-based sources are represented by operating system audit trail, system logs, and application-related audit data.
- *The taxonomy, which classifies data sources due to processing level*, consists of primary (raw), preprocessed, and generalized sources. *Primary* sources are network traffic, host command (system calls) traffic, and data from other sources. *Preprocessed* sources are *tcpdump* (for packets), preprocessed OS audit trail, system logs, and audit data of different applications. *Generalized* sources are generated by statistical processing of preprocessed sources.
- *The taxonomy, which classifies data sources due to an object, the data are associated with*, comprises network-based sources (packets, connections, all network traffic) and host-based

sources (traffic within a connection, processes, users, files and directories, disks, system registry, etc.).

The data sources that are supposed to be used in IDL case study are as follows:

- Network-based sources:
    - Preprocessed *tcpdump* data for IP, TCP, UDP, ICMP packets, and
    - *Tcpdump* statistical data.
- Host-based sources:
    - Preprocessed OS audit trail and statistical OS audit data;
    - System logs (for example, log and statistical data of commands run by the users plus resource, log and statistical data of all login failures, log and statistical data of all user logins/logouts and system startups and shutdowns);
    - Application audit data (for example, FTP logs and FTP statistical data, TELNET logs and TELNET statistical data, mail logs and Mail statistical data, HTTP logs and HTTP statistical data, DNS logs and DNS statistical data).

Four *typical structures* of data that can be used in IDL task:

- Time-based sequential data,
- Sequential (ordered) data,
- Relational (non-sequential) data, and
- Transactional data.

The *typical measurement scales* of ID learning data are as follows:

- Binary (or Boolean),
- Categorical,
- Linear ordered, and
- Real.

4. Development of mathematical methods realizing learning functionalities of the generic agents.

The multitude of methods that covers the needs of this task includes:
(1) methods for combining decisions produced by base-level classifiers on the basis of different data sources containing fragments of information about status of host operation security, and
(2) data mining and knowledge discovery techniques used for training and testing base-level classifiers.

Two *methods used for combining decisions* are chosen and described in detail. They are meta-classification method and competence-based method with some modifications developed by the authors of this Project. At that the first method was also validated on the basis of KDD Cup-99 case study.

From many existing methods *of data mining and knowledge discovery* developed for different types of data structure three basic methods were selected. The selection is based on analysis of data structures that can be perceived or computed on a host with the purpose of analysis of this host security status. The selected methods are:

1. **FP-growth** (*Frequent pattern growth*) method of frequent patterns and association rules mining aiming at extraction of useful patterns from transactional (sequential) data. This method was proposed recently and outperforms all known methods of the same type. This conclusion is made based on theoretical analysis of its complexity and also on the basis of its software implementation carried out by participants of this Project. Particularly, this method was implemented as a component of the *Server of learning methods*. Experiments carried out proved high efficiency of the *FP-growth* approach and this was the reason to arrive at the conclusion that it is reasonable to use this method instead of widely used group of methods based on *Apriori* algorithm.
2. **VAM** (*Visual Analytical Mining*) aiming at mining rules and other kinds of pattern from numerical data. This method was developed in depth and validated over several data sets from UCI repository. It is already implemented as a component of *Server of learning methods*.
3. **GK2** algorithm aiming at mining discrete data that was proposed, developed, implemented and validated by authors of this Report. The method is theoretically sound and showed good

efficiency. The advantage of this method is that it can also be used for extraction rules from data with missing values and at that it does not require to use a prognosis of missed values.

Case study "KDDCup-99" was used for the validation of both latter algorithms, and also multi-agent architecture of IDL system and implementation technology.

5. Development of the protocols of inter-level intelligent agent interaction (negotiation).

The following types of *the protocols of inter-level intelligent agent interaction (negotiation)* were developed:

- protocols for operation with particular data sources;
- protocols for creation of global coherent problem ontology, shared and private components of application ontology;
- protocols for combining decisions of source-based classifiers.

The most complex protocols are those for the task of creation of global coherent problem ontology, shared and private components of application ontology. This task consists in creation and synchronization of the source-based fragments of application ontology and their synchronization with the Data Fusion (DF) problem ontology.

These protocols serve for interaction of the IDLS entities located on different hosts during the design of the draft version of the application ontology and its iterative modification in the process of providing coherency. We have determined the protocols for creating the initial (basic) version of the application ontology (we called it the meta-protocols) and the protocols for further synchronization of the ontology during its iterative coordination with the local components of the ontology, as well as during any modifications to it.

We considered two meta-protocols: “top-down” and “bottom-up”. In the first case the expert of the meta-level responsible for forming the global ontology forms its basic variant that includes the list of basic application entities with the minimal necessary set of attributes and specifies the identifiers of entities. In the case of using multi-agent architecture of IDLS, special agent ("KDD master") managed by the meta-level expert sends out the basic variant of local fragments of the application ontology to the respective agents situated on the local data sources ("Data source managing agents" – DMAs) for analysis, correction, further expansion and filling. The DMAs of local sources managed by experts conduct modifications and extensions of the received ontology version aiming at the whole ontology coherency providing. The synchronization of changes and extensions of the first and next versions of ontology made by agents of the data sources is conducted by the meta-level agent step by step via exchange of messages with data source agents. The contents of the synchronization protocol consists in multi-phase negotiations, at that each source agent negotiates only regarding the shared and its own private part of the application ontology. These negotiations are carried out with the KDD master and result in the development of the application ontology which is coherent with problem ontology and has no contradictions at the level of application. All these procedures are performed under the supervision and with the active participation of meta-level expert and local source experts, interacting through their agents. After processing the above information, the local source DMA prepares proposals as to modifying and/or expanding the local components of the domain ontology and sends these proposals to the KDD master.

In the “bottom-up” protocol, local source experts first form the basic variants of the application ontology with regard to its shared part and their own private one, and then KDD master under supervision of meta-level expert performs the merging, coordination and correction of the received components of the application ontology to prepare the next basic variant of it. After that, the corresponding parts of this variant are sent to the local sources DMAs for further corrections if necessary. The subsequent work is performed in the manner similar to the above step of protocol.

In both protocols, the central component is its part that deals with the synchronization of the application ontology components proposed by KDD master and DMAs of the local sources of the application ontology.

In considering the interactions between the IDLS components we have to account for the possible spatial distribution of data sources and a possibility of non-reliable communication channels between the data sources and the meta-level host-server. For the realization of interaction mechanisms working under such conditions, protocols utilizing the two-phase lazy transactions have been used. These protocols are essentially similar to the database synchronization protocols, except that in the used synchronization protocols the verification of modifications carried out on the meta-level server is impossible. The overriding decision-making right in the area of modifications to the ontology resides with the meta-level application experts who bear the main responsibility for forming and supporting the global application ontology. Their responsibilities also include periodic review and verification of the modifications to the ontology proposed by the application experts who work with the local data sources.

6. Development of the technique for generalization of the particular agent decisions according to the meta-classification approach.

The developed technique for generalization of the particular agent decisions is based on a specified hierarchy of interaction of particular classifiers in process of producing a global decision on the basis of hierarchical combining of decisions of lower level classifiers.

Several basic approaches to combining decisions of multiple base-level classifiers have been analyzed. They can be grouped into four groups:

1. Voting algorithms;
2. Probability-based or fuzzy algorithms;
3. Meta-learning (meta-classification) algorithms based on stacked generalization idea;
4. Meta-learning algorithms based on classifiers' competence evaluation.

The meta-classification and competence-based methods were used and adapted in the Intrusion Detection Learning System (IDLS) under development. It is worth mentioning that both above methods cannot be used in "straightforward" manner because of peculiarities of data. Thus, these methods were adapted for application of distributed learning and decision making procedures. Again, most of these peculiarities are of technological character.

7. Development of architecture of the Multi-agent Learning System.

The developed architecture of the Intrusion Detection Learning System consists of the local data source components and the meta-level components.

The following main agents were included in the Intrusion Detection Learning System:

- *Intrusion detection KDD master agent*, which realizes distributed intrusion detection learning application ontology design, design of meta-model of decision making on intrusion detection, and distributed learning management.
- *Meta-level intrusion detection KDD agent* for distributed learning management, design of meta-model of decision making on intrusion detection, sending the decision making structures to local level agents;
- *Intrusion detection agent-classifier of meta-level* realizing distributed learning management and decision making on intrusion detection.
- *Data fusion management agent* intended for design of meta-model of decision making on intrusion detection, distributed learning management and decision making.
- *Intrusion detection KDD agent of a source* fulfilling design of meta-model of decision making and distributed learning management.
- *Intrusion detection agent-classifier of data source* executing distributed learning management and decision making.
- *Data source management agent* for decision making on intrusion detection, distributed application ontology design, design of meta-model of decision making, distributed learning management and data source monitoring to detect receipt of new data.

8. Development of object-oriented conceptual project of the Multi-agent Learning System.

Object-oriented project of IDLS was developed in terms of *Uses cases* diagrams, *collaboration* diagrams, *state-chart* diagrams and *component* diagrams.

Object-oriented project of IDLS includes the following specifications:

- the high level behavior of IDLS.
- the Base Classifier's decision making on intrusion detection.
- the Meta-classifier's decision making on intrusion detection.

- the IDLS agents' behavior in the processes of data preparing, informative features search, getting meta-properties of data, classifier learning, Meta-classifier learning, measurement scale transformation, construction of basic model of decision making knowledge base, basic classifier validation procedure, firing particular rule of Base Classifier knowledge base, design of meta-model of decision making, distributed application ontology design; etc.

On the whole, they provide for the necessary specifications of components of the system needed to write software code.

2.9. Current technical status

The progress in research fully matches the Work program and does not need any refinement.

2.10. Cooperation with foreign partners

According to the Work plan, one Interim Report (Interim Report # 2)was submitted to the Partner (by June 1, 2002). It contains the results of all predefined research. The Project executors together with the Partner representatives participated in the workshop, organized by the Partner to discuss the research results of the first year in March 2002 at Binghamton University, USA.

2.11. Problems encountered and suggestions to remedy

None

2.12. Perspectives of future developments of the research/technology developed

These perspectives will be discussed at the meeting that is planned to be held in April 2003. Proposal for continuation of the research supposed by this Project is submitted to Partner in July 2002.

Attachment 1: Illustrations attached to the main text

None

Attachment 2: Other Information, supplements to the main text

Brief content of the Interim reports submitted to the Partner

Interim Report #2

Attachment 3: Abstracts of papers and reports published during the year of reference

1. Gorodetski V., Karsayev O., Kotenko I., Khabalov A. Software Development Kit for Multi-agent Systems Design and Implementation // Lecture Notes in Artificial Intelligence, Vol.2296, Springer Verlag, 2002. P.121-130.

Abstract. The paper presents the developed technology and software tool for design and implementation of knowledge-based multi-agent systems. The software tool comprises two components that are "*Generic Agent*" and "*Multi-agent System Development Kit*" (MAS DK). The former comprises reusable Visual C++ and Java classes and generic data and knowledge base structures, whereas the latter comprises several developer-friendly editors aimed at formal specification of the applied multi-agent system (MAS) under development and installation of the resulting application in particular computer network environment. The developed technology and MAS DK were used in the design and implementation of the MAS prototype for computer network assurance and intrusion detection and distributed attack simulator.

2. Gorodetski V., Kotenko I. The Multi-agent Systems for Computer Network Security Assurance: frameworks and case studies // IEEE ICAIS-02. IEEE International Conference "Artificial Intelligence Systems". Proceedings. IEEE Computer Society. 2002. P.297-302.

Abstract. The paper presents experience in application of multi-agent technology for design and implementation of multi-agent systems (MASs) intended to cooperatively solve the currently critical tasks in the area of computer network security assurance. These MASs are Agent-based Simulator of Attacks against Computer Networks, Multi-agent Intrusion Detection System and Multi-agent Intrusion Detection Learning System. Each of these MASs is based on strict formal frameworks proposed by authors and designed and implemented as software prototypes on the basis of common technology and software tool "Multi-agent System Development Kit" developed by authors. The paper sketches the above MASs and analyses advantages of use of multi-agent architecture for computer network assurance.

3. V.Gorodetski. Multi-agent Data Fusion: Design and Implementation Issues. 5th International Conference on Information Fusion (Fusion-2002), CD Proceedings of the section "AFOSR Information Fusion Initiative". Annapolis, MD, USA, July 8-10, 2002.

Abstract. The following aspects of DF system design and implementation will be highlighted:
(1) How we resolve the data non-congruency problem caused by heterogeneity and distribution of data sources. In particular, how we provide for monosematic understanding of the terminology used in formal specification of distributed entities, how we solve entity identification problem, how we cope with the diversity of data physical nature, scales of measurement, variety of data accuracy, duplication of the same attributes in different data sources, and other.

(2) How we solve the task of distributed learning of DF system, i.e. what structures are used to combine particular decisions made on the basis of local data sources to generate global decision and what formal techniques are used to combine decisions in accordance with the hierarchy of classifiers.

(3) What is an architecture of multi-agent DF system and what new functionalities and respective agents (as compared with similar systems in data mining and knowledge discovery) should be used in this architecture.

In addition, we will present outline of a technology used for design and implementation of DF software tool. This technology is supported by the developed Multi-Agent System Development Kit (MAS DK) intended for the design and implementation of broad spectrum of multi-agent applications. In conclusion a brief outline of DF applications developed and being developed will be given.

4. V.Gorodetski, O.Karsayev and V.Samoilov. Multi-agent Data Fusion Systems: Design and Implementation Issues. Proceedings of the 10th International Conference on Telecommunication Systems - Modeling and Analysis, Monterey, CA, October 3-6, vol.2, pp.762-774, 2002.

Abstract. The objective of Data Fusion (DF) task is making decisions on the basis of distributed data sources accessible through Intra- or Internet. These sources contain heterogeneous data that can be represented in various structures (relational, transactional, etc.), can be of different nature (images, signals, numbers, etc), and accuracy, be measured in different scales (Boolean, categorical, real), can contain uncertainty, etc. An objective of a DF system is to combine data from many different sources to make decision, for instance, classification of an object or an object state, a situation assessment, etc. Within DF specific tasks three issues are of the most significance. The first is development of meta-model of distributed data sources; the second is development of meta-model of combining decisions produced on the basis of particular data sources and the third concerns architectural issue. According to our opinion, the first and the second tasks can be solved successfully only if we focus on the thorough development of the distributed application ontology playing a basic role in effective solution of many problems entailed by heterogeneity and distribution of data. In turn, the ontology-related solution significantly correlates to the architecture of the respective software. The above tasks and respective software tool considered from the design and implementation viewpoints are in the focus of the paper.

5. I.V.Kotenko. Multi-agent Technologies for Support of Intrusion Detection in Computer Networks. X Russian Conference "Methods and tools of information assurance". Proceedings. Saint-Petersburg, SPbSPU. 2002. P.44-45. (in Russian)

Abstract. The paper presents the basic applications of multi-agent systems in the field of intrusion detection in computer networks developed in Intelligent Systems Laboratory of SPIIRAS: (1) agent-based simulator of attacks on computer networks; (2) multi-agent intrusion (attack) detection system; (3) multi-agent system for intrusion detection learning in computer networks. Each of the developed applications is founded on the formal models and architectures offered by the scientists of the laboratory, and is realized by use of own toolkit for multi-agent system development MAS DK (“Multi-Agent System Development Kit”).

6. I.V.Kotenko. Case-based Recovering of Formal Grammars specifying Scenarios of Computer Attacks. Third International Conference "Artificial Intelligence -2002". Proceedings. 2002. Crimea, Ukraine, 2002. (in Russian)

Abstract. The paper analyzes the following approaches for synthesis of the formal grammars, specifying the scripts of computer network attacks: (1) inductive recovery on a set of precedents by means of formal methods; (2) definition by the expert on the basis of knowledge of malefactor’s intentions and possible ways of its realization; (3) combination of the first and second ways. The examples of the attack grammar recovery algorithms are submitted.

7. I.V.Kotenko. Case-based Recovering of Formal Grammars specifying Scenarios of Computer Attacks. International magazine “Artificial Intelligence”, № 3, 2002. (in Russian)

Abstract. In the paper, the approach for case-based synthesis (generation) of formal grammars specifying models of attacks against computer networks is considered. Two groups of grammar recovery algorithms are selected: enumeration grammar recovery algorithms and induction grammar recovery algorithms. The inductive method for

recovering regular grammars by a positive example (the Feldman method) has been chosen as the most appropriate. The description of its algorithmic implementation has been included. In order to demonstrate the performance capabilities of grammar recovery algorithms used to describe attacks on computer networks, several cases of computer network attacks are reviewed. The example of using grammar recovery algorithm for specification of computer network attacks is elaborated. This approach is intended for realization of multi-agent system of computer network attacks simulation.

8. I.V.Kotenko, O.V.Karsayev, V.V.Samoilov. Ontology of learning for Intrusion Detection in Computer Networks. International Conference on Soft Computing and Measurements. SMC'2002. Proceedings. Saint-Petersburg, 2002. Vol.1. P.255-258. (in Russian)

   Abstract. According to the modern view on the information system technology, ontology is one of the most important components of every information system, in particular, if the information system is highly distributed large scale and knowledge-based. The paper describes the developed architecture of Multi-agent Intrusion Detection Learning System and ontology of Intrusion Detection Learning domain intended for building this system. The ontology forms the high-level conceptual model of the basic shared knowledge represented as the structured set of basic notions with clearly and undoubtedly defined semantics independent from accepted design and implementation issues of a particular application. The specific of the subject domain is that it combines knowledge and therefore, ontologies, from three domains, namely, “Data Fusion and Data Fusion Learning problem domain ontology”, “Intrusion Detection subject domain ontology” and “Intrusion Detection Learning subject domain ontology”. These ontologies are described in the paper.

Attachment 4: Information on patents and property rights.

None

SIGNATURES

Task #2 Principal Investigator
Professor Igor V.Kotenko,
Doctor of Sciences (Tech), Ph. D.