ISTC 1994P

Project № 1994P

Annual Technical Report

For the first year
(December 01, 2000 - November 30, 2001)

1. Title of project

Formal Methods for Information Protection Technology

2. Contracting Institute

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)

3. Participating Institutes

None

4. Project Manager

Oleg V. Karsayev
(812)-323-3570,
(812)-328-0685,
ok@mail.iias.spb.su

5. Commencement Date, Duration

December 01, 2000
3 years

6. Partner

European Office of Aerospace Research and Development

SPIIRAS Director,

Prof. Rafael M.Yusupov

Project Manager

Dr. Oleg V. Karsayev

December 2001

Task # 1

FORMAL GRAMMAR-BASED FRAMEWORK, MODEL AND SOFTWARE TOOL PROTOTYPE FOR SIMULATION OF DISTRIBUTED ATTACKS ON COMPUTER NETWORK

1. Task Principal Investigator, phone number, fax number, e-mail address

Prof. Vladimir I. Gorodetski, (812)-323-3570, (812)-328-0685, gor@mail.iias.spb.su

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

Brief description of the work plan

A-1. Development of the specification of the representative set of distributed attacks defined on the macro-level.	1-3 Quarters
A-2. Development and selection of mathematical methods and techniques realizing the attack modeling.	2-3 Quarters
Interim Report #1 summarizing the efforts of the tasks A.1.	End of 2 Quarters
A-3. Development of the object-oriented project of the Attack Simulator–software tool prototype for simulation of attacks on the computer network	4-6 Quarter
Interim Report #2, summarizing the efforts of the tasks A.1 and A-2.	End of 4 Quarter
A-3. Development of the object-oriented project of the Attack Simulator–software tool prototype for simulation of attacks on the computer network	4-6 Quarter
Submission a paper in an International Journal	5 Quarter
Interim Report #2 documenting in brief the object-oriented project of the Attack Simulator software.	6 Quarter
A-4. Development of the software prototype of the Attack Simulator implementing theoretical results of research and its evaluation.	6-9 Quarters
Demonstration of the software components that will be used in the Attack Simulator software. (On demand of US AFRL/ID.)	9 Quarter
Final Report	9 Quarter

Notice: The grey shaded rows correspond to the tasks to be solved during the first year research. The Task A-3 had to be solved partially.

Objective

The main objective of the Task 1 of the project is the development of a formal model and software tool prototype for simulation of a broad versatility of distributed attacks and also exploration of their advantages as applied to the computer network assurance problems.

Expected results

The main expected result will be a formal framework, model, architecture and software tool prototype for simulation of distributed attacks on computer networks. In more detail, these results consist of the following:

1. Scenario-based specification of the representative set of distributed attacks (description of a set for particular cases of attacks specified on the macro-level);
2. Techniques for case-based regenerating (inductive recovery) of the formal grammar specifying formal model of the attack of the given class;
3. Stochastic model of a fragment of the attack at the micro level;

4. Object-oriented project of the Attack Simulator–software tool prototype for simulation of attacks on the computer network;

5. Attack Simulator software prototype, results of exploration of the developed Attack Simulator and evaluation of its advantages in the network assurance system design and maintenance.

Technical approach

A distributed attack is planned at a macro-level as a partially ordered set of steps forming an attack scenario. Each step aims at achieving a particular sub-goal corresponding to a particular "simple" attack. While implementing a distributed attack, at some steps the malefactor may succeed or fail. Each step can be implemented in many different ways. The same steps can be implemented in various orders and can be repeatable. They can be initialized from different computers and targeted against different computer network resources. Each particular attack can be implemented with various sequences of commands. In other words, the diversity of attacks and ways of their implementation can be formidable.

To reflect the above mentioned peculiarities of attacks the following technical approach is used within the task of interest.

At the higher (macro-) level, the attack scenarios are formalized in terms of a structured set of formal grammars interconnected with the "grammar substitution" operations. Each attack realization is specified as sequence of steps at the macro-level. Each such a sequence is interpreted as a "word" of a formal language specified formally in terms of a formal grammar. The set of "words" can be used as a training sample for inductive recovery of such a grammar. On the other hand, it is possible to use expert-based approach to formal grammars recovery and exactly this approach has already been employed. This way considered to be the most appropriate one because of a deficiency of representative samples of the representative diversity of attacks. The analysis performed and the results obtained proved that this way of action is appropriate. The analysis of the complexity of the grammars able to adequately specify the attacks against computer network also proved that one can restrict himself by usage of grammars which are not more complex than attributive (stochastic) *LL(2)*-grammars.

At the lower (micro-) level each step of an attack is detailed in terms of a sequence of events (system calls, OS commands, etc.). An event of this sequence realizes a particular action of a malefactor. In fact, this level of attack model details can also be formalized in terms of formal grammars via substituting terminals by sequences of commands. Other frameworks can also be used, e.g., hidden Markov model. But the latter formalism will be a subject of the research at the following phases.

3. Technical progress during the year of reference

Technical progress during the year of reference is fully compliant regarding both the tasks predefined in the Work plan and the schedule of their completion.

Achievements of the past year

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

1. Development of the specification of the representative set of distributed attacks defined on the macro-level.

2. Development and selection of mathematical methods and techniques realizing the attack modeling.

3. Development of the object-oriented project of the Attack Simulator–software tool prototype for simulation of attacks on the computer network.

The main results obtained during first year research are presented below.

1. The analysis and classification of the known attacks against single computers and also against computer network on the whole were carried out. A large number of computer attacks were analyzed. The taxonomy of attacks which was used as a basis to form the representative set of attacks to be modeled and simulated was developed.

2. The conceptual description of the representative set of the basic attack classes was developed. These attacks are considered as the components of distributed attacks forming their scenarios. The

scenario-based models of eight network attack classes are determined. These classes are the followings:

(1) Analysis of the network traffic,

(2) A network scanning (probing),

(3) A substitution of the trusted object of the network and transmission of the messages in its name with appropriation of its access rights,

(4) An implantation of the false object in a network,

(5) A denial of service,

(6) An unauthorized access from a remote computer by guessing password,

(7) An unauthorized access to local superuser (root) privileges, and

(8) A remote initiation of applications.

Each scenario is described by a set of admissible sequences of steps determining an attack class at macro and micro levels. Each attack class scenario is illustrated by a lot of particular examples specialized via concretization of the attacked computer specific software.

The models of the "*analysis of the network traffic*" attacks include a sequence of the following stages: determination of the places in the network where from the network should be listened to; determination of the analyzed OSI levels of network protocols and the protocol types; determination of the active network equipment over the network and mechanisms of its functioning; determination of software for an analysis and OS managing this analysis; adjustment of software and the development of rules (patterns) to filter basic information; analysis and selection of host masking means, when an intruder analyzes the traffic; implantation in the network and running the software (both analyzing the network traffic, and masking the intruder); gaining and analysis (filtering) of the traffic through the intruder's network; disconnecting from a network; analysis, decoding, and classification of information received by the intruder.

The most important stages, which can be presented in *network scanning,* are the following: selection of an "agent" computer and connection with it; determination of computers comprising the target network; recognition of the target network structure; recognition of the services being run on the target computer; getting additional information about the target network.

The common stages of the attack "*substitution of the trusted object of the network*" are: the preparatory stage, which is concerned with an analysis of the attacked objects and substitution of information on the server; listening to the network; sending a query (a storm of queries); sending a reply, computation of the next message number and its sending to the attacked host, rerouting the query to the intruder's host; execution of commands on the victim host; receiving and analysis of intercepted information; affecting the intercepted information; transfer of intercepted information (probably changed or substituted); propagation of the attack to other objects.

The models of the *"implantation of the false object into the network"* attacks include the following *stages*: studying the attacked network segment; listening to the network; sending a false message (or a storm of messages); receiving and analysis of intercepted information by the intruder or the "deceived" server; influencing on intercepted information; transferring the intercepted information (perhaps changed or substituted).

The models of the "*denial of service*" attacks contain a sequence of the following generalized stages: a reconnaissance of the network; an installation of master-agents and daemon-agents on the intermediary (auxiliary) hosts; sending messages from daemon-agents to master-agents (e.g., on a status); sending information on a status of daemon-agents from master-agents to a malefactor; sending commands from the malefactor's host to master-agents; sending commands from master-agents to daemon-agents; sending a specially crafted packet from the malefactor's host (or from the host used by the malefactor) to the intermediary host (or a set of intermediary hosts); the intermediary host (or a set of intermediary hosts) receives the packet and responds by sending a packet to the target host; the target host receives the packet and responds back to the intermediary host; sending a specially crafted packet (a sequence of the packets, fragments of the packet) from the malefactor's host (from the host used by the malefactor or by daemon-agents) to the target host.

The main stages of the "*unauthorized access from a remote machine by guessing password*" attacks are: getting information about the target system and its authentication subsystem; getting information about users of the target system; an interception of ciphered (or hashed) passwords; getting database with ciphered (hashed) passwords; single entering of the password in online mode; a multiple entering of passwords in online mode; a retrieval of the passwords in offline mode; an interception of passwords in text format (may be used for some other services).

The following stages are similar for the "*unauthorized access to local superuser (root) privileges*" attacks: an analysis of the attack targets; a preparation of the code; an implantation of the code; an implantation of parameters (parameterization of the code); a transfer of control to the code.

The "*remote initiation of an application*" attacks are characterized by the following stages: a reconnaissance of the target computer system; an implementation of a malicious code or program text into the target system; an unauthorized access to the system resources; an initiation and use of auxiliary software, legally installed in the target system; an initiation of a malicious program; an activation of some special functions, available in the implemented malicious program; sending information from the implemented program to the intruder; cleaning log files and deleting other attack evidences; a self-reproduction of a malicious program.

3. The thorough study of formal frameworks that can potentially be used as a formal mechanism for distributed attack modeling and subsequent simulation proved that formal grammar framework matches the attack modeling domain in the best way. Formal grammar mathematical framework is used for formal specification of real objects or sets of objects of a regular structure. This conclusion is also applicable to the task of specification of scenarios of complex network attacks. Therefore, the scenario of an attack against a computer system may be adequately represented in terms of formal grammar or a family of nested grammars. Besides, this grammar may play a dual role, namely in may be used both as a model of attack cases generation, and as a model of attack detection based on a syntax analyzer that corresponds to a grammar used in the model formal specification.

4. A thorough analysis of the methods of synthesis (recovery) of grammars applied to attack modeling task was carried out. Formally, the task of grammar synthesis consists in creating the algorithm to recover its syntactic structure based on the finite set of words of the language that specifies attack cases, and on the finite set of words from the supplement to this language. The grammar that generates scenarios for computer network attacks can be synthesized: (1) through inductive recovery based on the set of cases through the use of formal methods; (2) by an expert who possesses knowledge of the malicious party's intentions and the possible ways these intentions can be realized; (3) through combining the two above methods.

Two groups of algorithms are selected for grammar recovery: (1) enumeration grammar recovery algorithms; (2) induction grammar recovery algorithms. The inductive grammar recovery methods are deemed the most adequate to recover grammars that specify computer network attacks; particularly, the inductive method for recovering regular grammars on the basis of positive examples (the Feldman method). This method consists in constructing a non-recursive grammar that creates precisely those strings that were present in the training example, and then arriving at a simpler recursive grammar that generates all the strings of the positive examples and an infinite amount of other strings.

In order to verify the performance capabilities of the above algorithms several cases of computer network attacks have been analyzed. These cases set the basic methods of implementing attacks of the following types: network scanning for identification of hosts; network scanning for identification of services; identification of operating system; shared resource enumeration; users and groups enumeration; applications and banners enumeration; actions on getting access to resources; denial of service attacks.

The examples of using grammar recovery algorithms for specification of computer network attacks were developed. These examples have proved the practical applicability of the algorithms represented here for the specification of computer network attacks. The synthesized grammars can generate the versions of attacks that served as the training data for the development of the grammar. The grammars developed through combining a number of productions can generate the attacks not included in the training cases. This expands the capability of the attack simulator based on the utilization of these grammars.

5. A multi-layer formal model of the representative set of distributed attacks specified in terms of a family of interacting context-free attributed grammars is developed. Such a model allows to specify and to simulate distributed attacks on various levels of details. The formal model includes specifications of all its basic components. They are as follows:

- *Basic notions and components of distributed attack specification*. They include specifications of scenarios of distributed attacks and malefactors' intentions. In the developed model the malefactor's intention-centric approach to the specification of its activity is used. This means that basic notions of the domain correspond to the malefactor intentions and all other notions are structured according to the structure of intentions.
- *Ontology of the "Network Attacks" domain*. It interacts with the family of the aforementioned formal grammars, and each notion of the ontology corresponds to a symbol of the sequence specifying attack scenario.
- Formal model of the attacked computer network. In this model the attacked network is considered as environment that reacts to the malefactors' actions. *The main attributes* taken into account in the host model are: *IP*-address, mask of the network address, type and version of OS, users' identifiers, their domain names, passwords, users' security identifiers (*SID*), domain parameters, active ports of the hosts (services used, busy *TCP* and *UDP* ports, etc.), running applications, security indices, shared resources, trusted hosts and some other attributes.

Based on an implementation issue, the distributed attack is considered as a sequence of coordinated actions of spatially distributed malefactors. The developed architecture of the attack simulator implementing the above described attack model is being built as a multi-agent system (MAS). Each malefactor is represented as an intelligent agent of the same architecture and possessing similar functionality. While developing an attack, they interact via message exchange informing each other about current state and results of the attack in order to coordinate their further activity. These messages represented in standard MAS communication languages, i.e. in KQML, that is standard of DARPA (for message "wrapper"), and XML (for message content). The design and implementation of the attack simulator in being carried out on the basis of the Agent System Development Kit that was developed by the authors.

6. During the reported period the partial object-oriented design of the macro-level components of the Attack simulator is fulfilled.

The specifications of the following components are developed:

(1) Model of the malefactor’s actions;
(2) Model of the computer network under attack and the host models;
(3) Model for calculating the probability of successful actions (attacks) against the host;
(4) Model of host response to malefactor’s actions.

According to Work plan the object-oriented design of the rest of the components of the attack simulator must be completed by the end of Q6.

4. Current technical status

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

5. Cooperation with foreign partners

According to the Work plan, two Interim Reports were submitted to the Partner (before June 1, 2001 and before December 1, 2001). They contain the results of all predefined research.

The Project executors together with the Partner representative participated in the European Summer School on Multi-agent-systems. In March of 2001 the project leader attended the Partner institution in the USA in order to discuss the forthcoming research. Partner's representative attended the St. Petersburg Institute for Informatics and Automation in May of 2001. In March of 2002 the Partner is going to organize a workshop in the USA aimed at a discussion of the interim results and further research on the Project

6. Problems encountered and suggestions to remedy

None

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

These perspectives will be discussed during the March of 2002 meeting in the USA.

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 #1

Interim Report #2

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

1. V.Gorodetski, I.Kotenko, Models of attacks against computer networks based on usage of formal grammars. *Proceedings of the International conference on soft computing.* St. Petersburg, Russia, pp. 212-216, 2001 (in Russian).

   Abstract. The paper describes formal models of attacks against computer network based on grammars. An analysis of computer network attacks is elaborated. A two level conceptual model of computer network attacks (on macro and micro-level) is presented. At the first (macro) level the common attack scenario is represented as an ordered set of actions. The second (micro) level determines more detailed specification of attack as operating system commands, standard applications, and exploits using concrete call parameters. The formal models of attacks at each level are set with the use of stochastic context free grammars. This model determines partially ordered sequences of activities at each level in the form of a set of specifications of the respective attacks. Each such a sequence can be considered as a "word" belonging to a formal language that, in turn, is specified by a formal grammar.

2. V.Gorodetski, O.Karsaev, I.Kotenko, A.Khabalov. MAS DK: Software Tool for Multi-agent Systems Design and Examples of Applications. *Proceedings of the International Congress "Artificial Intelligence in XX1 Century"* (ICAI 2001), Russia, September 3–8, 2001, Physmatgis Publishers, Moscow, Russia, pp. 249-262, 2001 (in Russian).

   Abstract. In the paper, the developed technology and respective software tool aiming at design and implementation of multi-agent system is described. This software tool comprises a multitude of reusable components assembled together within a so-called "Generic Agent". The design and implementation of an applied multi-agent system is realized as the process of specialization of the Generic agent classes and data structures. This process is supported by a so-called "Multi-agent System Development Kit". The resulting specification of the system in question is presented in the "System Kernel", which is a specialized collection of databases, reusable classes and specialized library. The paper also describes three multi-agent applications that have already been or are being currently developed. They are (1) MAS for operation planning; (2) MAS for data mining and knowledge discovery; and (3) Multi-agent intrusion detection system.

3. V.Gorodetski, O.Karsaev, I.Kotenko, A.Khabalov. Software Development Kit for Multi-agent Systems Design and Implementation. *International Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAS-2001),* Krakow, Poland, September 2001 (the paper also will be published in "Lecture Notes in Artificial Intelligence" series of Springer Verlag Publishers in 2002).

   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, intrusion detection, and distributed attack simulator. Several other applications are currently under development.

4. V.Gorodetski, I.Kotenko, Man'kov E.V. Simulation of distributed attacks against computer networks. *Proceedings of the II-d Interregional conference on Information security of Regions of Russia, (IBBR-01*), St. Petersburg, Russia, November 26-28 2001, pp.56-57 (in Russian).

   Abstract. The paper considers the developed formal model of distributed attack built as a multi-level hierarchy of stochastic attribute context-free grammars interconnected via "formal grammar substitution" operation. The conceptual basis of the model is formed by the domain ontology developed by authors. Conceptually, each node of the ontology is mapped onto a goal, which a team of malefactors is aiming to achieve, or a particular action. In the software tool, the developed grammars are simulated by the interacting set of state machines. Software tool also includes a model of the victim computer network, within which a security policy is implemented and its reaction at the malefactor's actions (commands). Attack simulator is implemented as a multi-agent system, where each malefactor is represented by a software agent, and computer network model is considered as an environment. While performing an attack, malefactors communicate via messages exchange specified in terms of KQML and XML languages.

Attachment 4: Information on patents and property rights.

Task manager
Ph.D. Prof. V.Gorodetski