AGENT-BASED MODELING AND SIMULATION OF CYBER-WARFARE BETWEEN MALEFACTORS AND SECURITY AGENTS IN INTERNET

Igor Kotenko
St. Petersburg Institute for Informatics and Automation
39, 14th Liniya, St. Petersburg, 199178, Russia
E-mail: ivkote@iias.spb.su

KEYWORDS
Agent-based Modeling and Simulation, Computer network attacks, Information Assurance.

ABSTRACT

The paper considers an approach to modeling and simulation of cyber-wars in Internet between the teams of software agents. Each team is a community of agents cloned on various network hosts. The approach is considered by an example of modeling and simulation of "Distributed Denial of Service" (DDoS) attacks and protection against them. Agents of different teams compete to reach antagonistic intentions. Agents of the same team cooperate to realize joint intentions. The ontologies of DDoS-attacks and mechanisms of protection against them are described. The variants of agents' team structures, the mechanisms of their interaction and coordination, the specifications of hierarchy of action plans as well as the developed software prototypes are determined.

1. INTRODUCTION

The modern society greatly depends on various distributed computer systems, which have different possibilities and are characterized by high complexity. Vulnerabilities of these systems, permanently magnified quantity, variety and complexity of cyber-attacks and gravity of their consequences highlight urgent necessity for information assurance and survivability of computer systems. Especially it is fair in connection with integration of computer systems on the basis of open networks (such as the Internet), permanently modified and magnified, not having state boundaries, centralized control and uniform security policy. Hackers characterize the current state of counteraction of malefactors' systems to security systems as "a game of network cats and mice" (Nomad 2002).

Experienced malefactors realize sophisticated strategies of cyber-attacks. These strategies can include:

(1) Information gathering about the computer system under attack, detecting its vulnerabilities and defense mechanisms;

(2) Determining the ways of overcoming defense mechanisms (for example, by simulating these mechanisms);

(3) Suppression, detour or deceit of protection components (for example, by using slow ("stretched" in time) stealthy probes, separate coordinated operations (attacks) from several sources formed complex multiphase attack, etc.);

(4) Getting access to resources, escalating privilege, and implementation of thread intended (violation of confidentiality, integrity, availability, etc.) using the vulnerabilities detected;

(5) Covering tracks of malefactors' presence and creating back doors in order to use them later.

Protection mechanisms should support real-time fulfillment of the following operations:

(1) Implementing the protection mechanisms appropriated to the security policy (including proactive intrusion prevention and attack blocking, misinformation, concealment, camouflage, etc.);

(2) Vulnerability assessment, gathering data and analysis of the current status of the computer system defended;

(3) Intrusion detection and prediction of the malefactors' intentions and actions;

(4) Direct incident response, including deception of the malefactors, their decoy with the purpose of disclosure and more precise determining the malefactors' purposes, and reinforcement of critical protection mechanisms;

(5) Elimination of intrusion consequences and detected vulnerabilities, adaptation of the information assurance system to the next intrusions.

Unfortunately, the existing theoretical base for information assurance in large-scale systems does not correspond to the indicated tendencies.

We think that the majority of problems in information assurance is caused by immaturity of logical foundations for construction of integrated adaptive security systems operating in adversarial environments (Ellison et al. 1997; Linger et al. 2001).

To our opinion, it is stipulated mainly by insufficient attention to fundamental works, which, on the one hand, consider information assurance as a complex task of organizational and technical cyber warfare between security systems and malefactors' attacking systems (Bell and Santos 2002; Carmel and Markovitch 1996; Garstka 2000; Geib and Goldman 2001; Goldman 2002; Modeling and Simulation Activities 1997), and, on the other hand, are based on *exploratory modeling and simulation* of indicated processes (Klein 2002; Stytz and Banks 2001; Yuill at al. 2000).

Proceedings 19th European Conference on Modelling and Simulation
Yuri Merkuryev, Richard Zobel, Eugène Kerckhoffs © ECMS, 2005
ISBN 1-84233-112-4 (Set) / ISBN 1-84233-113-2 (CD)

The issues of modeling and simulation of information assurance are actively researched during more than thirty years. The various formal and informal models of particular protection mechanisms are developed, but practically there are not enough works formalizing complex antagonistic character of information assurance.

This state of research is explained by complexity of information security problems (Evans et al. 2001; Faatz 2000; Waag et al. 2000).

Understanding of an information assurance as uniform holistic system is extremely hampered. It depends on great many interactions between different cyber warfare processes and is determined by dynamic character of these processes and different components of computer systems (Bell and Santos 2002).

Especially it is fair in conditions of the Internet evolution to a free decentralized distributed environment in which a huge number of cooperating and antagonistic software components (agents) interchange among themselves and with people by large information contents and services (Information Dynamics and Emergent Behavior 2001; Kephart at al. 1998). Modeling and simulation of these aspects of information assurance is supposed to put as a basis of our research. This will allow developing an integrated approach to construction of network security systems operating in aggressive antagonistic environment.

Information assurance modeling and simulation is considered in the paper as modeling and simulation of processes realized, at least, by two opposing sides, namely, malefactors' attacking and security systems trying to bypass, deceive and suppress each other. Such modeling and simulation can allow developing main principles of construction of intrusion-aware distributed network security systems, which operate by prediction of intentions and actions of malefactors.

One of the most harmful classes of attacks aiming at destruction of network resources availability is "Denial of Service" (DoS). The purpose of DoS is isolation of a victim host, i.e. creation of a situation in which a remote host cannot communicate with an external world. The basic feature of "Distributed Denial of Service" (DDoS) attacks is coordinated use of enormous remote hosts-"zombies" for generation of ill-intentioned traffic (Mirkovic et al 2002; Noureldien 2002; Jeon 2001). DDoS attacks are preceded by breakings of tens (and sometimes hundreds, and even thousand) computers in which the special DDoS-software is established thus allowing to carry out coordinated DoS attacks against victim hosts.

The goal of our research is development of mathematical basis for adaptive co-evolving agent-based adversarial modeling and simulation for cyber warfare. This paper considers an approach to agent-based modeling and simulation of cyber warfare by an example of modeling and simulation of DDoS attacks (Gorodetsky at al. 2003) and protection against them.

The rest of the paper is structured as follows. *Section 2* outlines suggested common approach for modeling and simulation of cyber warfare by imitating counteraction of malefactors and defense agents' teams. *Section 3* describes the ontology of DDoS attacks and defense mechanisms. *Section 4* presents specifications of structure and scheme of operation of DDoS agents' team. *Section 5* outlines structure and functioning of defense agents' team. *Section 6* depicts mechanisms for action coordination, monitoring and restoration of agent functionality, and maintenance of communication selectivity. *Section 7* describes the suggested formal model of computer network. *Section 8* determines architecture and main user interfaces for software prototypes developed. *Conclusion* outlines the results of the paper.

2. TEAMWORK-BASED APPROACH FOR MODELING AND SIMULATION

Agent-based modeling and simulation of information assurance assumes that cyber warfare is represented as a large collection of semi-autonomous interacting agents. The aggregate system behavior emerges from evolving local interactions of agents in a dynamically changing environment specified by computer network model.

We assume to select two agents' subsystems (teams) effecting on computer network as interconnected set of objects (resources):

(1) Adversary attacking system - a team of malefactor's agents (for automatic generation of distributed coordinated attacks);

(2) Security (defense) system - a team of security agents (for intrusion protection, data sensing and information fusion, intrusion detection, adversary intentions and actions prediction, and incident response).

So the task of agent-based modeling and simulation for information assurance is represented as modeling and simulation of the malefactor and security agents-teams' interaction including the security (defense) team response on adversary activity.

Thus, adversary and defense systems are represented as antagonistic teams of agents. The purpose of adversary team consists in defining vulnerabilities of computer network and defense system and implementation of security threats (confidentiality, integrity, availability, etc.) by executing distributed coordinated attacks. The purpose of defense team is protection of computer network and own components from attacks.

Agents of different teams compete to reach opposite intentions. Agents of the same team cooperate to achieve common intention (to fulfill attack on computer network or to defense the computer network). Competing agents must deal with probabilistic knowledge, risk management, deception, and opponent modeling.

The security agents protect the defended network from attacks, observe adversary's actions, fulfill information fusion, try to deceive adversary team, estimate enemy actions, status of the network and components of the

security system, predict malefactors' intentions and actions, react to enemy actions, restore resources of the network (Gorodetski et al. 2004).

It is said that the agents' team realizes teamwork, if the team members (agents) fulfill joint operations for reaching the common long-time goal in a dynamic external environment at presence of noise and counteraction of opponents. Now the research on teamwork is an area of steadfast attention in multi-agent systems. A set of approaches to formalization and simulation of the agents' teamwork is known (Cohen and Levesque 1991; Grosz and Kraus 1996; Jennings 1995; Tambe 1997; Tambe and Pynadath 2001; etc).

For implementing teamwork, we use the main ideas stated in works on the joint intention theory (Cohen and Levesque 1991), the shared plans theory (Grosz and Kraus 1996) and the combined theories of agents' teamwork (Jennings 1995; Tambe 1997; Tambe and Pynadath 2001; etc).

In our approach it is offered that the agents' teamwork is organized by the group (team) plan of the agents' actions. In result, a team has a mechanism of decision-making about who will execute particular operations. As in the joint intention theory, the basic elements, allowing the agents' team to fulfill a common task, are common (group) intentions, but its structuring is carried out in the same way as the plans are structured in the shared plans theory (Kotenko 2003). The common (group, individual) intention and commitment are associated with each node of a general hierarchical plan. These intention and commitment manage execution of a general plan, providing necessary flexibility. During functioning each agent should possess the group beliefs concerning other team-mates. For achievement of the common beliefs at formation and disbandment of the common intentions the agents should communicate. All agents' communications are managed by means of common commitments built in the common intentions. For this purpose it is supposed to use the special mechanism for reasoning of agents on communications. Besides it is supposed, that agents communicate only when there can be an inconsistency of their actions (Tambe 1997). It is important for reaction to unexpected changes of network environment, redistributing roles of the agents which failed or unable to execute the general plan, and also at occurrence of not planned actions. The mechanisms of the agents' interaction and coordination can be based on three groups of procedures (Tambe and Pynadath 2001):

(1) Coordination of the agents' actions (for implementation of the coordinated initialization and termination of the common scenario actions);

(2) Monitoring and restoring the agents' functionality;

(3) Communication selectivity support (for choice of the most "useful" communication acts).

The specification of the plan hierarchy is carried out for each role. The following elements of the plan should be described: initial conditions, when the plan is offered for fulfillment; conditions for finishing the plan execution (these conditions can be as follows: plan is fulfilled, plan is impracticable or plan is irrelevant); actions fulfilled at the team level as a part of the common plan. For the group plans it is necessary to express joint activity.

Assignment of roles and allocation of plans between the agents is fulfilled in two stages: at first the plan is arranged in terms of roles, and then the roles are put in correspondence to the agents. Agents' functionalities are generated automatically according to the roles specified.

The adversary (malefactors') team co-evolves by generation of new attack patterns to overcome defenses. On the other hand, defense team co-evolves by generating new protective actions against attacks, suppression of malefactors' team and recovery of destructed and compromised components of the computer network.

Interaction among agents can be represented as a two-player game ("game of network cats and mice"), where the agents' objective is to look for a strategy that maximizes their expected sum of rewards in the game.

We assume that agents' strategies can be modeled by means of the family of stochastic attribute formal grammars (and their interpretation by state machines) and hidden markov models.

To cope with the information heterogeneity and distribution of intrusion sources and agents used we apply ontology-based approach and special protocols for specification of shared consistent terminology.

The suggested technology for creation of the malefactors-agents' team (that is fair for other subject domains) consists in realization of the following chain of stages (Gorodetsky, at al. 2003; Kotenko 2003):

(1) Formation of the subject domain ontology;

(2) Determination of the agents' team structure;

(3) Determination of agents' interaction-and-coordination mechanisms (including roles and scenarios for roles exchanges);

(4) Specification of agents' plans as a hierarchy of stochastic formal grammars;

(5) Assignment of roles and allocation of plans between agents;

(6) State-machine based implementation of teamwork.

Modeling in any subject domain assumes development of its conceptual model, i.e. a set of basic concepts of the subject domain, relations between the concepts, and also data and algorithms interpreting these concepts and relations. So formation of the subject domain ontology is an initial stage of the agents' team creation.

The agents' team structure is described in terms of a hierarchy of group and individual roles. Leaves of the hierarchy correspond to the roles of individual agents, but intermediate nodes - to group roles. One agent can execute a set of roles. Agents can exchange roles in progress of plan execution.

For agents' team operation in real-time a hierarchy of state machines is used. These state machines are built as a result of interpretation of a hierarchy of attribute

stochastic formal grammars which set the plan hierarchy specification. The state machines implement a choice of the plan which will be executed and a fulfillment of the established sub-plans in a cycle "agents' actions - environment responses".

Agents' coordination is carried out by message exchange. As the agents' teams operate in antagonistic environment agents can fail. The lost functionalities are restored by redistributing the roles of failed agents between other agents and (or) cloning new agents.

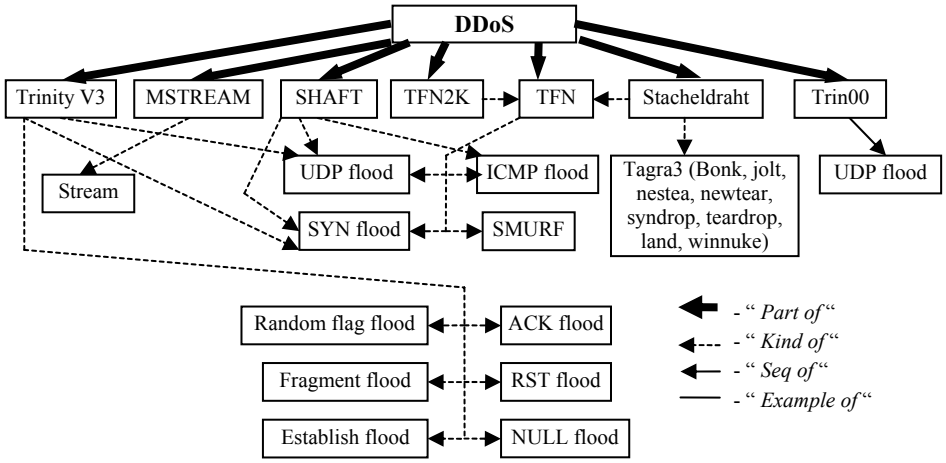

Figure 1: Fragment of DDoS Attacks Ontology on Macro-level

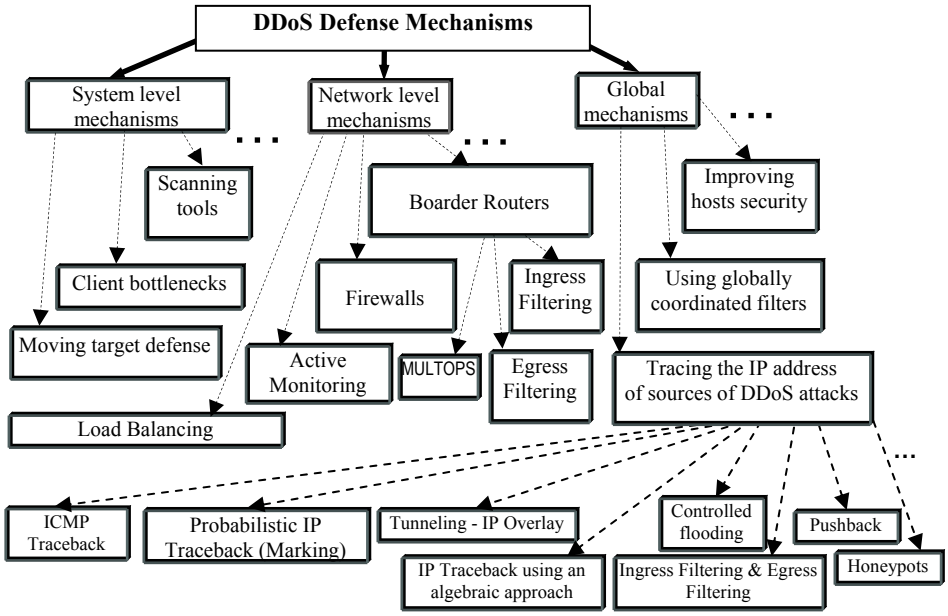

Figure 2: Fragment of DDoS Defense Ontology on Macro-level

3. ONTOLOGY OF ATTACKS AND DEFENSE MECHANISMS

The developed ontology comprises a hierarchy of notions specifying activities of team of malefactors and defense agents directed to implementation of DDoS attacks and protection from them in different layers of detail. In this ontology, the hierarchy of nodes representing notions splits into two subsets according to the macro- and micro-layers of the domain specifications. All nodes of the ontology on the macro- and micro-levels of specification are divided into the intermediate (detailable) and terminal (non-detailable) (Gorodetski and Kotenko 2002).

The ontology notions of an upper layer can be interconnected with the corresponding notions of the lower layer through one of three kinds of relationships: (1) "Part of" that is decomposition relationship ("Whole"–"Part"); (2) "Kind of" that is specialization relationship ("Notion"–"Particular kind of notion"); and (3) "Seq of" that is relationship specifying sequence of operation ("Whole operation" – "Sub-operation").

High-layer notions (corresponding to the intentions) form the upper layers of the ontology. They are interconnected by the "Part of" relationship. Attack and defense actions realizing agent's intentions (they presented at lower layers as compared with the intentions) are interconnected with the intentions by "Kind of" or "Seq of" relationship.

The "terminal" notions of the macro-level are further elaborated on the micro-level, and on this level they belong to the set of top-level notions detailed through the use of the three relationships introduced above.

In micro specifications of the ontology, besides three relations described ("Part of", "Kind of", "Seq of"), the relationship "Example of" is also used. It serves to establish the "type of object–specific sample of object" relationship.

The developed ontology includes the detailed description of the domain of DDoS attacks and defense mechanisms in which the notions of the bottom layer ("terminals") can be specified in terms of network packets, OS calls, audit data, etc.

The fragment of the DDoS-attacks sub-ontology is depicted in Figure 1. The nodes specifying a set of software exploits for generation of DDoS attacks

(Trinity V3, MSTREAM, SHAFT, TFN2K, Stacheldraht, Trin00) make up a top level of the ontology fragment (Figure 1). At lower levels of the fragment different classes of DoS-attacks are detailed. For example, "Smurf" attacks consist in sending broadcasting ICMP ECHO inquiries on behalf of a victim host, therefore hosts accepted such broadcasting packages answer to the victim host, that results in essential capacity reduction of a communication channel and, in some cases, in full isolation of an attacked network. Other example is "Land" attack. It is sending an IP-packet with equal fields of port and address of the sender and the receiver, i.e. Source Address (SA) = Destination Address (DA), Source Port Number (SPN) = Destination Port Number (DPN).

The nodes determining the high levels of defense mechanisms (system, network, global) make up the top level of DDoS defense mechanisms ontology (Figure 2). At the bottom levels of the ontology these nodes are described by particular defense mechanisms (Mirkovic at al. 2002; Noureldien 2002).

Different types of nodes corresponding to system level defense mechanisms can be used. For example, scanning tools check presence of DDoS-agents in the host file system, and also scan the ports frequently used by attackers. Mechanisms of client bottlenecks are directed on creating bottleneck processes on the zombie hosts used for DDoS-attacks to limit their attacking ability. Mechanisms of moving target defense consist in changing IP address to avoid being attacked.

Mechanisms of a network level can be represented by the following nodes: (1) mechanisms for defeating DDoS attacks at boarder routers (Ingress Filtering, Egress Filtering, MULTOPS bandwidth attack detection, etc.); (2) firewalls (intended for eliminating the packets implementing DoS-attacks); (3) active monitoring (for continuous supervision on network state, checking TCP/IP traffic and reaction to critical situations); (4) load balancing (on the basis of configurable input and output queues on hosts).

Global mechanisms implement coordinated defense tools on different hosts in the Internet. For example, the goal of tracing the source IP address is to observe the intruders' path back to the zombie computers or the original attacker.

4. ATTACK AGENTS' TEAM

We limit the *team of DDoS agents* by three-level structure. The team consists of the "client" supervising a subteam of "masters". Each master, in turn, manages a group of "demons". "Demons" execute immediate attack actions against victim hosts. "Demons" include two subsets – scouts D_R and attackers D_A.

So a set of agents can be described as a pair $A = \{M, D\}$, where $M=\{m_1, m_2, ..., m_r\}$ – set of "masters"; $D=\{D_R, D_A\}=\{d_1, d_2, ... , d_n\}$ – set of "demons" ("scouts" and "attackers"). Each agent can be represented as follows (Georgeff and Rao 1995; Rao and Georgeff 1998): $a_N = <K, B, De, I, P, G, C>$, where

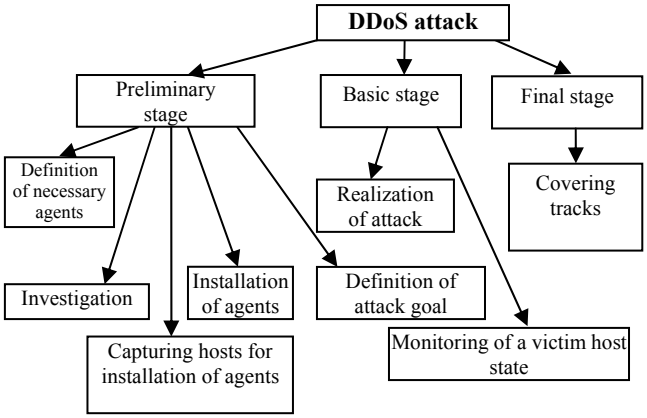

Figure 3: Upper level of DDoS Agents' Plans Hierarchy

N – an identifier of the agent; K – knowledge; B – beliefs; De – desires; I – intentions; P – a set of parameters determining a mode of agent's operation, for example, minimal reaction time (RT), etc.; $G=\{L_R, f_R\}$ – a set of agent's goals and actions, L_R – hierarchy of possible goals and actions of the agent (reactions to influences), f_R – anytime-function for selecting a current goal or action from L_R according to K, B, De, I, P and C for time not exceeding RT; C – commitments of the agent concerning other agents.

Common formal plan of attacks implemented by team of DDoS agents has three-level structure: (1) Upper level is a level of intention-based scenarios of malefactors' team specified in terms of sequences of intentions and negotiation acts; (2) Middle level is a level of intention-based scenarios of each malefactor specified in terms of ordered sequences of sub-goals; (3) Lower level is a level of malefactor's intention realization specified in terms of sequences of low-level actions (commands).

The upper level of hierarchy of DDoS agent plans is depicted in Figure 3. DDoS-attack includes three stages: preliminary, basic and final. Main operations of the preliminary stage are investigation (reconnaissance) and installation of agents. The content of the basic stage is realization of DDoS attack by joint actions of agents. Having received as a result of a chain of messages a "victim" address, agents-attackers begin to defeat a chosen host. At this time agents-scouts monitor a victim state. At detection of success of attack agents-scouts inform other agents about this fact. In case of prevarication of a host (for example the host been switched off from a network) or impossibility of defeating it, the operation is terminated or a new victim for DDoS attack is chosen.

The fragment including the upper and middle levels of hierarchy of agent plans for preliminary and basic stages is depicted in Figure 4. As an example, only two types of DDoS-attacks are represented – SMURF and Land attacks.

Mathematical model of attacks is specified in terms of a set of formal grammars interconnected through "substitution" operations (Kotenko and Man'kov 2003):

M_A=<{G_i}, {Su}>, where {G_i} – the formal grammars, {Su} – the "substitution" operations. The sequences of symbols ("strings", "words" – in formal grammar terminology) generated by each of such grammars correspond to the sequences of time ordered malefactor's intentions and/or actions. It is assumed that every sequence of a malefactor's actions viewed as a "word" in a formal language is specified through a family of enclosed context-free grammars recognizable by a corresponding family of state machines. At the scenario specification layer (it was earlier called macro-layer) such sequences correspond to the specification of scenarios in terms of the malefactor's intentions and actions.

The formal model of attack scenarios in terms of formal grammars are based on the attacks ontology described above. It is noteworthy to notice that each node of the ontology that is not "terminal" one is mapped to particular grammar, which is capable to generate only admissible sequences realizing this intention in terms of symbols, corresponding to the ontology nodes of the immediately lower layer. Depending on the required level of detail, these nodes may be represented by the terminal nodes of the macro or micro-level. In the former case, the grammar may be used to visualize the malefactor's actions, and in the latter case – for attack simulation in the lowest layer terms (if the "terminal" nodes of the micro-level are represented by network packets, or messages of the tcpdump program, OS commands and/or calling applications with specified parameters).

Every formal grammar is specified by quintuple (Kotenko and Man'kov 2003): $G=<V_N,V_T,S,P,A>$, where G is the grammar identifier (name), V_N is the set of non-terminal symbols (that are associated with the upper and the intermediate levels of representation of the steps of an attack scenario), V_T is the set of its terminal symbols (that designate the steps of a lower-level attack scenario), $S \in V_N$ is the grammar axiom (an initial symbol of an attack scenario), P is the set of productions (production rules) that specify the refinement operations for the attack scenario through the substitution of the symbols of an upper-level node by the symbols of the lower-level nodes, and A is the set of attributes and algorithms of their computation.

Attribute component of each grammar serves for several purposes. The first of them is to specify randomized choice of a production at the current inference step if several productions have the equal left part non-terminals coinciding with the active non-terminal in the current sequence under inference.

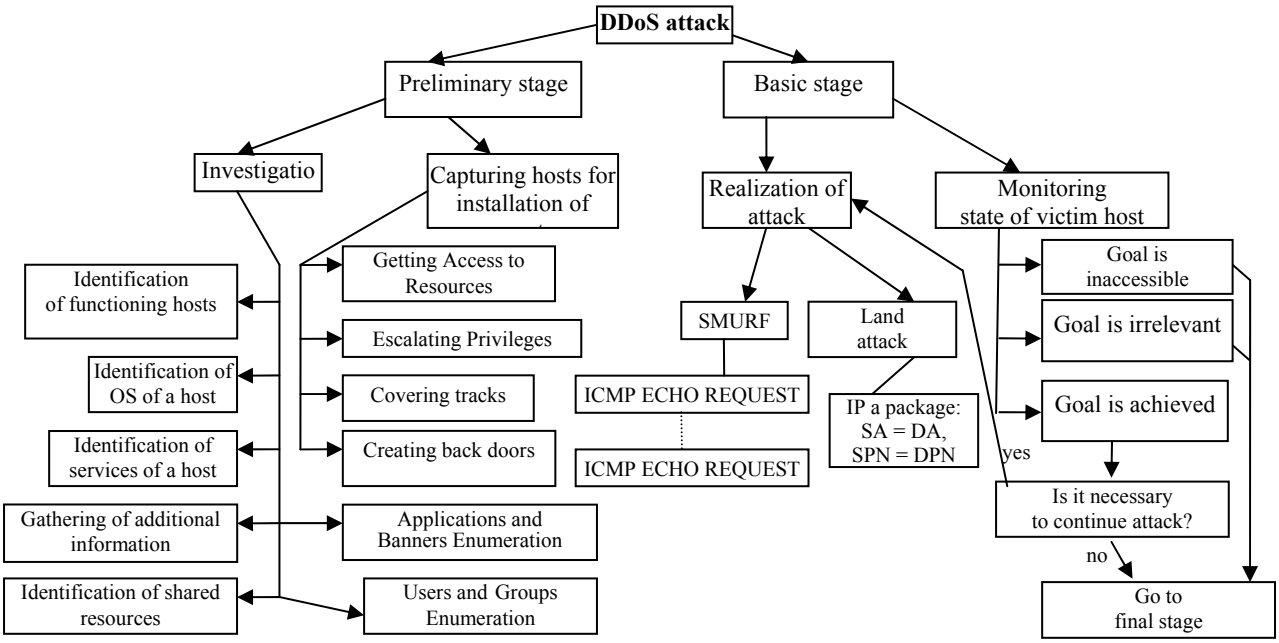

Figure 4: Fragment of Upper and Middle Levels of Hierarchy of DDoS Agents' Plans

Also the attribute component is used to check conditions determining the admissibility of using a production at the current step of inference. These conditions depend on (1) task specification (general attack goal), (2) configuration of the attacked computer network (host) and its resources and (3) results of the malefactor's previous actions.

Assignment of roles and distribution of plans between agents are carried out as follows: roles of the agents necessary for the given goal are selected, the chosen roles are appointed to agents, agents of corresponding types are installed (cloned, employed).

5. DEFENSE AGENTS' TEAM

Defense agents are installed on hosts placed in critical points of a network. The team of defense agents on a host can consist of one or several agents of various classes (Figure 5).

Sensor agent (SA) carries out preliminary processing of network traffic fixing events which are significant for

defense. *Identification and authentication agent (IAA)* is responsible for determining sources of messages and acknowledgement of their authenticity. *Access control agent (ACA)* regulates access of users to network resources according to their privileges. Agents *IAA* and *ACA* detect non-authorized actions on access to information resources of a host and interrupt connections attributed to number non-authorized. *Intrusion detection agents (IDA)* are responsible for detection of particular "suspicious" events or obvious attacks and decision-making concerning reaction to these events. *Intelligent intrusion detection agents (IIDA)* realize higher level of processing and generalization of facts discovered. They makes decision on the basis of messages about detected suspicious behavior and obvious attacks from different agents. *Reaction agents (RA)* are responsible for suppressing

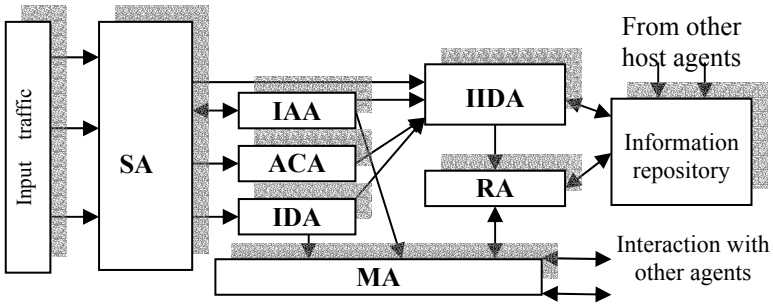

Figure 5: Structure of Defense Agents' Team

attacks. *Meta-agent (MA)* controls agents' behavior and detects complex distributed attacks which particular actions directed on different hosts.

The hierarchy of action plans of defense agents consists of three basic levels of processing:

(1) Agents *SA* fulfill primary processing of input traffic;

(2) Agents *IAA, ACA* and *IDA* carry out in real time the preliminary analysis of the received data detecting obvious attacks and reacting on them;

(3) Agents *IIDA* and *MA* detect and react to the multiphase distributed attacks. These agents also implement forecasting the subsequent actions of users, using known scripts of attacks.

6. INTERACTION AND COORDINATION MECHANISMS

As it was described in section 2, it is offered to use three groups of mechanisms (procedures): maintenance of action coordination, monitoring and restoration of agent functionality, and supporting communication selectivity (Tambe 1997).

The mechanisms of first class are intended for realization of coordinated initialization and termination of actions of some general plan. Coordinated initialization means that all team-mates begin execution of the same plan at defined time. It assumes an appointment of specific agents for fixed roles in specific scenario, their notification about the appointed scenario and role, and also reception of confirmations on their readiness to play the defined role in the scenario.

The coordinated termination of a common action (refusal of the common intention) demands also mutual informing of team-mates about the action at presence of corresponding conditions. Such conditions can be determined by achieving the common goal, finding-out of unattainability of the goal or its prevarication by even one member of the team.

For example, the attack goal "increase of authority up to a level of superuser" is achieved, if a malefactor managed to penetrate into a target host and to increase his authority up to a level of superuser. Also the purpose is unattainable, if one of obligatory actions on penetration into a target host is not executed. And the purpose is irrelevant, if the target host is switched off from the network.

Mechanisms of monitoring and restoration of agent functionality should provide supervision of some agents over others that it was possible to establish loss of capacity for work by the agent or a group of agents. It is directed on fast restoration of functionality of the team at the expense of reassignment of the "lost" roles to those team-mates which can perform corresponding job.

For example, if one of the malefactors-agents who are carrying out intention "Identification of operating system of a host" is blocked by firewall of a target network or other obstacle for realization of this intention takes place, this agent (or other malefactor-agent who found out state of nonoperability of "colleague") should send this information to a "leader" of the scenario. If there will be other agent, capable to solve the task this role should be assigned to it. Checking of rules and realization of reasoning should entail corresponding communications of agents by means of some communication protocol.

Mechanisms of maintaining communication selectivity order the communication act when the probability and cost of agents' coordination loss is great enough. They are based on calculating importance of the message in view of the "costs" and benefits of this message. It is necessary to guarantee that the benefit of the message exchange for maintenance of agents' coordination surpasses a "cost" of the communication act (for example, a network security system, having intercepted agents' messages, can detect and "suppress" an attack). Therefore it is very important to choose those communication acts which will bring the greatest benefit to the team.

7. MODEL OF COMPUTER NETWORK

The attack development depends on the malefactor's "skill", information regarding network characteristics, which he/she possesses, some other malefactor's attributes (Gorodetski and Kotenko 2002), security

policy of the attacked network, a power of the defense agents' team, etc.

An attack is being developed as interactive process, in which the attacked network and security agents are reacting on the malefactor's action. Computer network plays the role of the environment for attacker and security agents, and therefore its model must be a part of the attack simulation tool.

Model of the attacked computer network is represented as quadruple $MA = <M_{CN}, \{M_{Hi}\}, M_P, M_{HR}>$, where M_{CN} is the model of the computer network structure; $\{M_{Hi}\}$ are the models of the host resources; M_P is the model of computation of the attack success probabilities; M_{HR} is the model of the host reaction in response of attack. Let us determine the model M_{CN} of a computer network structure *CN* as follows: $M_{CN} = < A, P, N, C >$, where A is the network address; P is a family of protocols used (e.g., TCP/IP, FDDI, ATM, IPX, etc.); *N* is a set $\{CN_i\}$ of sub-networks and/or a set $\{H_i\}$ of hosts of the network *CN*; *C* is a set of connections between the sub-networks (hosts) established as a mapping matrix. If *N* establishes a set of sub-networks $\{CN_i\}$, then each sub-network CN_i can in turn be specified by the model M_{CNi} (if its structure needs to be developed in detail and if information is available about this structure). Each host H_i is determined as a pair $M_{Hi} = <A, T>$, where *A* is the host address, *T* is a host type (e.g., firewall, host, etc.).

Models $\{M_{Hi}\}$ of the network host resources serve for representing the host parameters that are important for attack simulation.

Success or failure of any attack action (corresponding to terminal level of the attack ontology) is determined by means of the model M_P of computation of the attack success probabilities. This model is specified as follows: $M_P = \{R^{SPr}_j\}$, where R^{SPr}_j is a special rule that determines the action success probability depending on the basic parameters of the host (attack target). The rule R^{SPr}_j includes IF and THEN parts. The IF part contains action name and precondition (values of attributes constraining the attack applicability). The THEN part contains value of success probability (*SPr*).

The result of each attack action is determined according to the model M_{HR} of the host reaction. This model is determined as a set of rules of the host reaction: $M_{HR} = \{R^{HR}_j\}$, R^{HR}_j: *Input* → *Output* [& *Post-Condition*]; where *Input* – the malefactor's activity, *Output* – the host reaction, *Post-Condition* – a change of the host state, & – logical operation "AND", [] – optional part of the rule. The Input format: *<Attack name>: <Input message> : <Attack objects>* [; *<Objects involved in the attack>*. The Output format: {*<Attack success parameter S>* [: *<Output message>*]; {*<Attack success parameter F>* [: *<Output message>*]}. The Attack Success Parameter is determined by the success probability of the attack that is associated with the host (attack target) depending on the implemented attack type. The values of attack success parameter are Success (S), and Failure (F). The part of output message shown in the < > is taken from the corresponding field of the host (target) parameters. The part of output message shown in quotation marks "" is displayed as a constant line. The Post-Condition format: $\{p_1=P_1, p_2=P_2, ..., p_n=P_n\}$, where p_i – i^{th} parameter of the host (for instance, *SP, SR, TH*, etc.) which value has changed, P_i – the value of i^{th} parameter.

8. PROTOTYPES AND THEIR EVALUATION

Using Java, Visual C ++ and MASDK (Gorodetski et al. 2002) several prototypes of particular components of multi-agent system intended for simulation cyber-war of agents' teams in computer networks were developed. We have implemented three software components: a component for simulation of DDoS-agents' operation, a component of simulation of DoS-attacks and a component of intrusion detection and reaction to DDoS-attacks. Now these components are used for validation of the accepted basic ideas and formal framework. These components allow to show all stages of DDoS-attacks in the evident form, and also to simulate different situations depending on security parameters of attacked networks. The model of agents' team has three-level structure consisting of "client" managing "masters" which supervise "demons" (Figure 6).

"Demons" are subdivided into two roles - scouts and attackers. Figure 7 illustrates capturing hosts and installing DDoS agents.

The visualization of particular DoS attacks is shown in Figure 8. The information presented in the figure is divided in four groups: (1) the attack task specification placed in the left top-most part of the screenshot; (2) the attack generation tree visualized in the right hand part of it; (3) the strings of the malefactor's actions placed in the left hand part of the screenshot and below the attack task specification; (4) a tag of success (failure) as green (black) quadrate and data obtained from an attacked host (a host response) depicted on the right hand part of each information warrior's action.

The main objective of the experiments with the prototypes is to evaluate the tool's efficiency for simulation of different DDoS attacks and attacked network configurations. These experiments were carried out for various parameters of the attack task specification and an attacked computer network configuration.

The influence of the following input parameters on attacks efficacy was explored: a malefactor's intention, a degree of protection afforded by the network and personal firewall, a degree of security of attacked host, and the degree of malefactor's knowledge about a network.

The following parameters of attack realization outcome have been selected: *NS* (Number of attack Steps) – number of terminal level attack actions; *PIR* (Percentage of Intention Realization) – percentage of the hacker's intentions realized successfully (for "Reconnaissance" it is a percentage of objects about which the information has been received; for

"Implantation and threat realization" it is a percentage of successful realizations of the common attack goal on all runs); *PAR* (Percentage of Attack Realization) – percentage of "positive" messages (responses) of the Network Agent on attack actions (the "positive" messages are designated in attack visualization window by green lines); *PFB* (Percentage of Firewall Blocking) – percentage of attack actions blockage by firewall (red lines in attack visualization window); *PRA* (Percentage of Reply Absence) – percentage of "negative" messages (responses) of the Network Agent on attack actions (gray lines in attack visualization window).

Let consider two generated dependences of parameters PIR, PAR, PFB, PRA from different input parameters values under intention GAR ("Gaining Access to Resources") realization. This intention is implemented on the stage "Capturing hosts for installation of agents". For construction of these dependences the following values of the attacked network configuration were used as x-coordinate parameters: 1 – both network and personal firewalls are active; 2 – only network firewall is active; 3 – only personal firewall is active; 4 – none of firewalls is active.

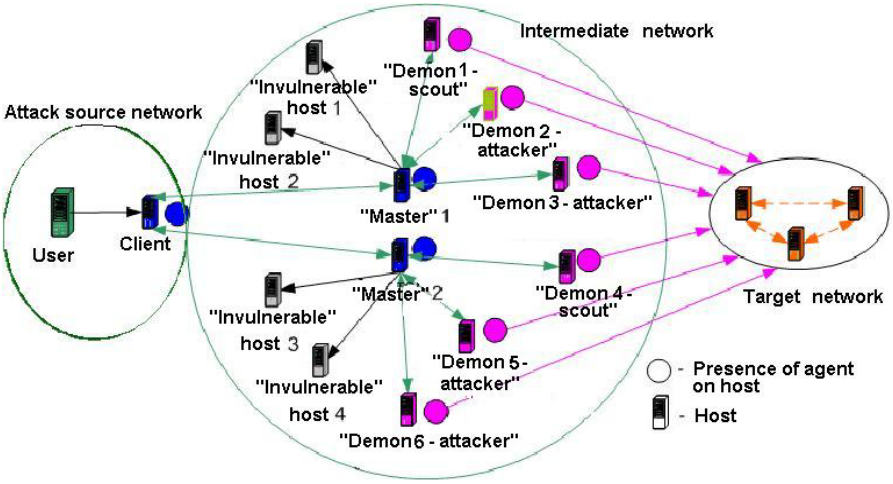

Figure 6: Diagram of DDoS Attacks Simulation

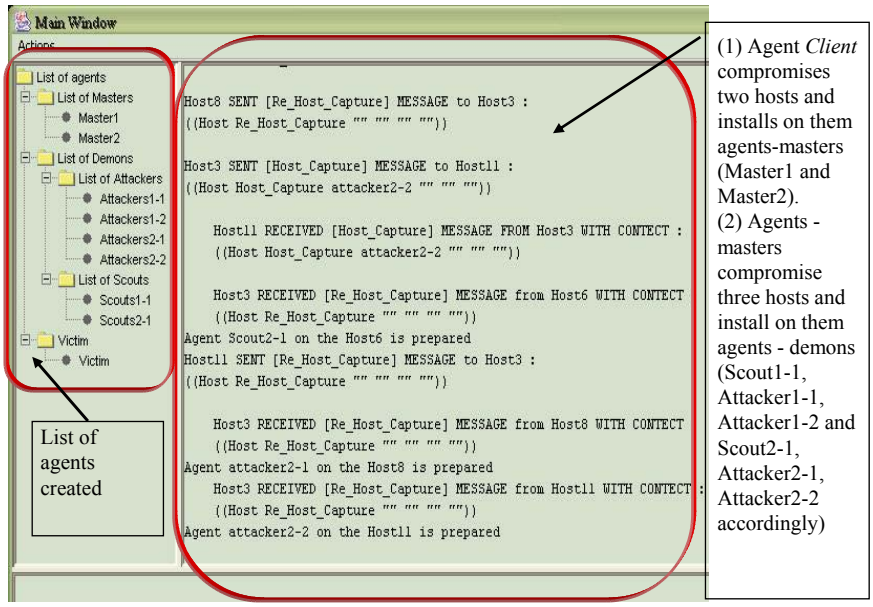

Figure 7: Capturing Hosts and Installing DDoS Agents

The main parameters changes under maximal protection of attacked host ("Strong") and maximal hacker's knowledge about a network ("Good") are depicted in Figure 9.

The main parameters changes under minimal protection of attacked host ("None") and maximal hacker's knowledge about a network ("Good") are depicted in Figure 10.

The simulation-based exploration has demonstrated its efficacy for accomplishing various attack scenarios against networks with different structures and security policies implemented.

9. CONCLUSION

In the paper we developed basic ideas of the modeling and simulation of cyber warfare by teamwork approach and formal grammars. The technology for creation of the agents' team was suggested and described. We developed the approach to be used for conducting experiments to both evaluate computer network security and analyze the efficiency and effectiveness of security policy against different DDoS attacks.

We presented the structure of a team of agents, specifications of hierarchies of agent plans, agent interaction-and-coordination mechanisms, and agent role-assignment mechanisms.

Software prototypes were developed. They allow imitating a wide spectrum of real life DDoS attacks and defense mechanisms. Software code is written in terms of Visual C++ 6.0 and Java 2 languages. Experiments with the prototypes have been conducted, including the investigation of attack scenarios against networks with different structures and security policies.

The further development of our modeling and simulation framework and software tools will consist of joining different software components implemented, improving capabilities of the attack and defense agents

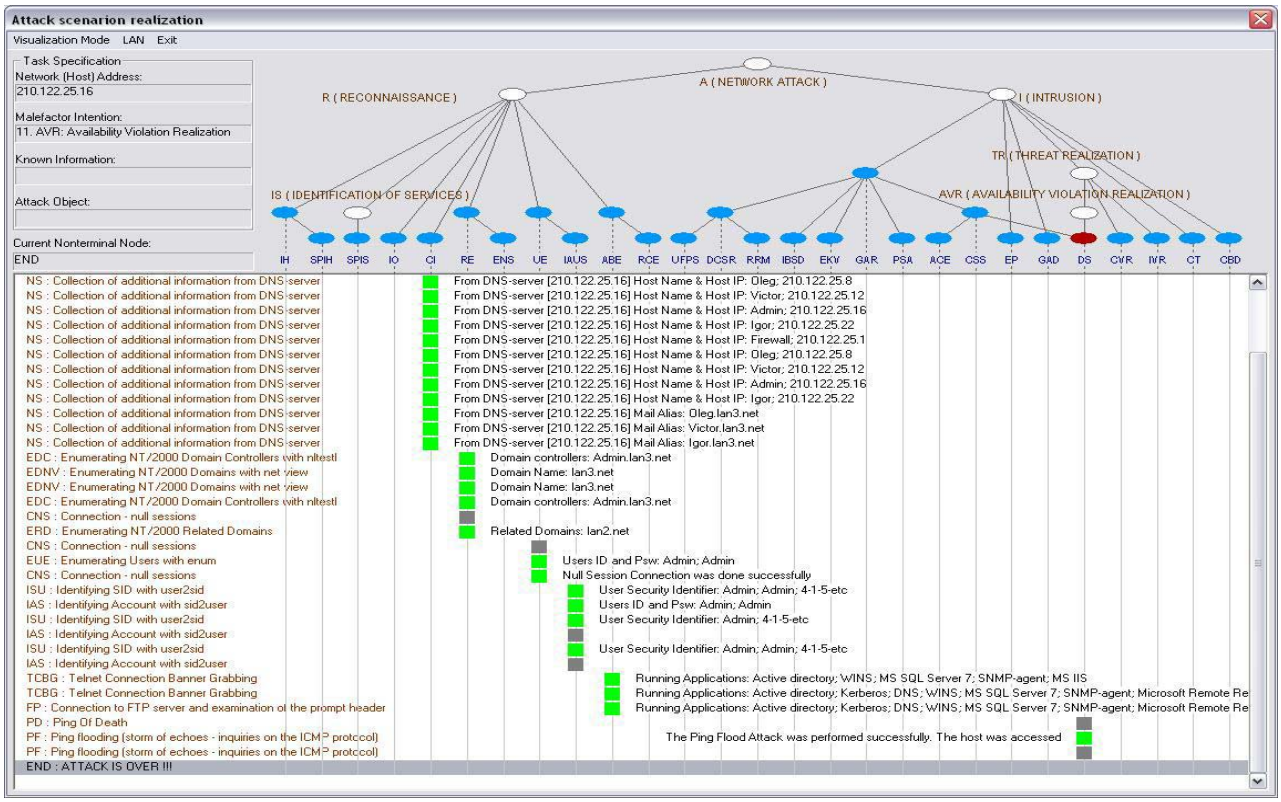

Figure 8: Visualization of DoS Attack Generation

teams by expansion of the attack and defense mechanisms classes, and implementing more sophisticated attack and defense scenarios.

10. ACKNOWLEDGEMENT

This research is being supported by grants 04-01-00167 of Russian Foundation of Basic Research and partly funded by the EC as part of the POSITIF project (contract IST-2002-002314).

REFERENCES

Bell, B. and E. Santos Jr. 2002. "Making Adversary Decision Modeling Tractable with Intent Inference and Information Fusion". *Proc. of the 11th Conf on Computer Generated Forces and Behavioral Representation*, Orlando FL.

Carmel, D. and S. Markovitch. 1996. "Opponent modeling in multi-agent systems". *Adaption and Learning in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence. Vol.1042.

Cohen, P.R. and H.J. Levesque. 1991. "Teamwork". *Nous*, 25(4).

Ellison, R.J.; D.A. Fisher; R.C. Linger; H.F. Lipson; T. Longstaff; N.R. Mead. 1997. *Survivable Network Systems: An Emerging Discipline*. Technical Report. November.

Evans, S.; S.F. Bush; J. Hershey. 2001. "Information Assurance through Kolmogorov Complexity". *DARPA Information Survivability Conference and Exposition II*. Anaheim, California.

Faatz, D. 2000. "Information System Security Assurance Evaluation". *Information Assurance Technology Symposium*.

Garstka, J. 2000. *Network Centric Warfare: An Overview of Emerging Theory*. PHALANX.

Geib, C.W. and R.P. Goldman. 2001. "Plan recognition in intrusion detection systems". *DARPA Information Survivability Conference and Exposition*, DARPA and the IEEE Computer Society.

Georgeff, M.P. and A.S. Rao. 1995. "The semantics of intention maintenance for rational agents". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada.

Goldman, R.P. 2002. "A Stochastic Model for Intrusions. Recent Advances in Intrusion Detection". *Fifth International Symposium. RAID 2002*. Proceedings. Lecture Notes in Computer Science, V.2516. Springer Verlag.

Gorodetski, V.; O. Karsayev; I. Kotenko; A. Khabalov. 2002. "Software Development Kit for Multi-agent Systems Design and Implementation". Lecture Notes in Artificial Intelligence, Vol. 2296.

Gorodetski, V. and I. Kotenko. 2002. "Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool". *Fifth International Symposium. RAID 2002*. Proceedings. Lecture Notes in Computer Science, V.2516. Springer Verlag.

Gorodetsky, V.I.; I.V. Kotenko; J.B. Michael. 2003. "Multi-agent Modeling and Simulation of Distributed Denial of Service Attacks on Computer Networks", *Third International Conference "NAVY AND SHIPBUILDING NOWADAYS"* (NSN'2003). Proceedings. St.Petersburg, Russia.

Gorodetski, V.; O. Karsayev; I. Kotenko, V.Samoilov. 2004. "Multi-Agent Information Fusion: Methodology, Architecture and Software Tool for Learning of Object and Situation Assessment". *The 7th International*

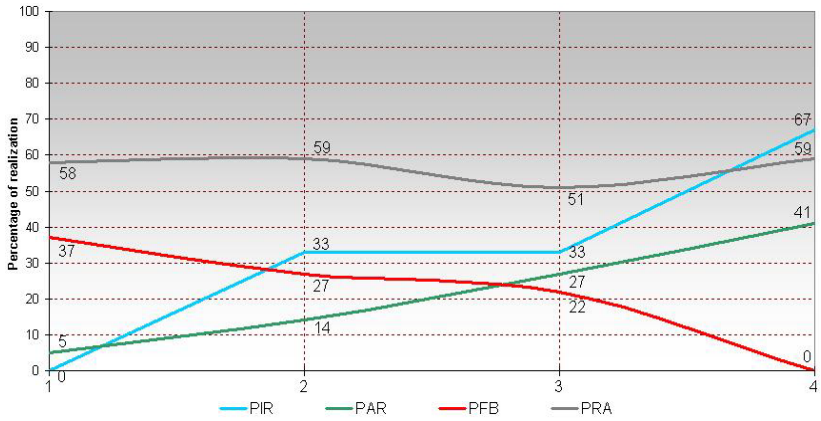

Figure 9: Changes of Simulation Parameters (1)

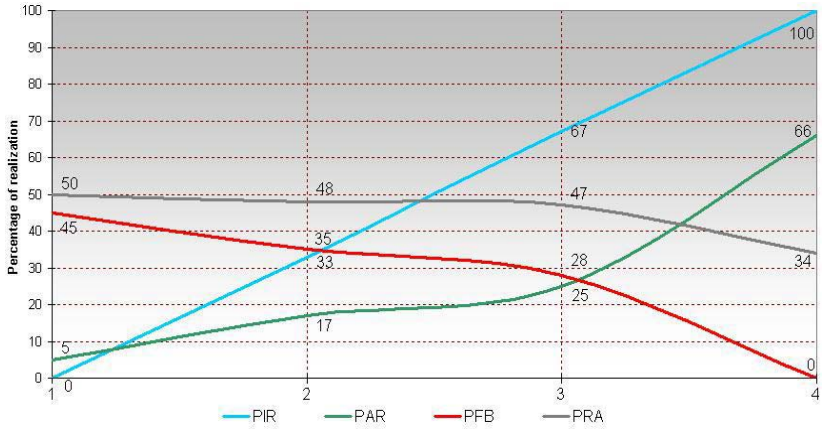

Figure 10: Changes of Simulation Parameters (2)

Conference on Information Fusion. Proceedings. Stockholm, Sweden. June 28 - July 1, P.346-353.

Grosz, B. and S. Kraus. 1996. "Collaborative plans for complex group actions". Artificial Intelligence, Vol.86.

Information Dynamics and Emergent Behavior. 2001. *Information Dynamics and Emergent Behavior of Heterogeneous-Agent Systems.* Arizona State University. DARPA ITO Sponsored Research. Project Summary**.**

Jennings, N. 1995. "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions". *Artificial Intelligence*, No.75.

Jeon, DeokJo. 2001. *Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation*. SANS Institute.

Kephart, J.O.; J.E. Hanson; J. Sairamesh. 1998. "Price-War Dynamics in a Free-Market Economy of Software Agents". *Proceedings of the Sixth International Conference on Artificial Life*. Editors MIT Press.

Kotenko, I. 2003. "Teamwork of Hackers-Agents: Modeling and Simulation of Coordinated Distributed Attacks on Computer Networks". *Lecture Notes in Artificial Intelligence*, Vol.2691.

Kotenko, I. and E. Man'kov. 2003. "Agent-Based Modeling and Simulation of Computer Network Attacks". *Fourth International Workshop "Agent-Based Simulation 4 (ABS 4)"*. Proceedings. Jean-Pierre Muller, Martina-M.Seidel (Editors). April 28-30. Montpellier. France.

Klein, G.L. 2002. *The Future for Intelligent Simulation Models*. http://www.mitre.org/pubs/ edge/january_02/ klein.htm

Linger, R.C. and A.P. Moore. 2001. *Foundations for Survivable System Development: Service Traces, Intrusion Traces, and Evaluation Models*. Report CMU/SEI-2001-TR-029.

Mirkovic, J.; J. Martin; P. Reiher. 2002. *A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms*, Technical report #020018. Computer Science Department, University of California, Los Angeles.

Modeling and Simulation Activities. 1997. *Modeling and Simulation Activities in Support of Information Assurance.* Report IATAC TR-97-002.

Nomad. 2002. *Nomad Mobile Research Centre*. www.nmrc.org.

Noureldien, A.N. 2002. "Protecting Web Servers from DoS/DDoS Flooding Attacks. A Technical Overview". *International Conference on Web-Management for International Organisations*. Proceedings, Geneva.

Rao, A. S. and M. Georgeff. 1998. Decision procedures of BDI logics. *Journal of Logic and Computation*, 8(3).

Stytz, M.R. and S.B. Banks. 2001. *Requirements and Issues in Cyberwarfare Simulation*. http://www.sisostds.org/doclib/ .

Tambe, M. 1997. "Towards Flexible Teamwork". *Journal of Artificial Intelligence Research*, No.7.

Tambe, M. and D.V. Pynadath. 2001. "Towards Heterogeneous Agent Teams". *Lecture Notes in Artificial Intelligence*, Vol.2086.

Waag, G.L.; J.M. Feinberg; L.J. Painchaud; R.K. Heist. 2000. *Information Assurance Modeling & Simulation. State of the Art Report - A Summary.* Modeling and Simulation Information Analysis Center, Alexandria. VA.

Yuill, J.; F. Wu; J. Settle; F. Gong; R. Forno; M. Huang; J. Asbery. 2000. Intrusion-detection for incident-response, using a military battlefield-intelligence process. *Computer Networks*, 34.

BIOGRAPHY

IGOR KOTENKO graduated with honors St.Petersburg Academy of Space Engineering (Department of Telecommunication and Computer-Aided Systems) and St.Petersburg Signal Academy (Department of Computer-Aided systems). He obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Leading researcher of St. Petersburg Institute for Informatics and Automation. He is now leading a research group in the field of multi-agent modeling and simulation for computer network security. His e-mail address is ivkote@iias.spb.su and his Web-page can be found at http://space.iias.spb.su/ai/kotenko/kotenko.jsp .