

Agent-based modeling and simulation of network softbots' competition

Igor KOTENKO¹ and Alexander ULANOV¹
SPIIRAS, Computer Security Research Group, Russia

Abstract. The research devoted to design and implementation of new knowledge-based methods and tools for verification and validation of complex software systems is now an important direction of scientific investigations. The paper describes the approach and software environment developed for agent-based modeling and simulation of defense against coordinated distributed attacks in the Internet. According to this approach, the cybernetic opposition of malefactors and defense systems in the Internet is represented by competition of antagonistic softbots' teams. The possibility of the approach application is analyzed by testing the defense mechanisms against Distribute Denial of Service attacks.

Keywords. Intelligent agents and softbots, knowledge-based methods and tools for testing, verification and validation, computer network security

Introduction

Nowadays one of very important research directions is connected with *development of new knowledge-based methods and tools for comprehensive investigation of complex systems (including testing, verification and validation of software components)*. It is especially true for the systems which operate in *distributed competitive noisy environments*, where the large groups of various defense and offense components counteract and collaborate (for example, in such domains as information warfare, competition in business, computer network assurance, etc.).

One of the solutions of this problem can be based on the *investigative modeling and simulation* using the family of various approaches (system dynamics, discrete-event and agent-based simulation, etc.) and different models (from analytical to scaled-down and full-scale). The choice of specific approaches and models depends on the goals of investigation, the complexity of subject domain, and the necessary fidelity and scalability of models. Our interest is related to the competition processes taking place in the Internet consisting in interaction of large number of cooperating and antagonistic software agents or softbots. Analytical models let imitate global processes (including viral epidemic), but describe the ones only on the abstract level. Packet-level simulation of network processes gives the opportunities to improve the fidelity of simulation, and can represent attack and defense actions as packet exchange. Those models can precisely specify these actions on the data link, network, transport and application layers. The greatest fidelity is archived with hardware testbeds. But these testbeds succeed in simulation of sufficiently limited fragments of interactions.

¹ St. Petersburg Institute for Informatics and Automation, 39, 14th Liniya, St. Petersburg, 199178, Russia;
E-mails: ivkote@iias.spb.su, ulanov@iias.spb.su

The paper describes an *integrated agent-oriented and packet-level approach* for simulation of competition processes in the Internet elaborated by authors. We suggest using this approach for investigation of distributed defense mechanisms which can be deployed in the Internet for protection from distributed coordinated network attacks. The possibility of the approach application is analyzed by testing the defense mechanisms against one of the most dangerous classes of network attacks – DDoS (Distribute Denial of Service) [22]. According to the approach suggested, the cybernetic counteraction of “bad guys” and security systems is represented by the *interaction of different softbots’ teams*. The aggregated system behavior becomes apparent by means of the local interactions of particular softbots in dynamic environment that is defined by the model of computer network. We distinguish at least the team of malefactors and the defense team. The softbots from the same team collaborate to achieve the joint intention (to realize the threat or to defend the network).

The main basis for the research is the agent teamwork theory. There are three well-known approaches to the formalization of the agent teamwork – joint intentions theory [5], shared plans theory [12] and the hybrid approaches [14, 26] which use the combination of joint intentions and shared plans theories. A lot of teamwork approaches are implemented in various multi-agent software, e.g. GRATE*, OAA, CAST, RETSINA-MAS, COGNET/BATON, Team-Soar, etc.

Another fundamental component of the research is represented by the studies on reasoning systems about opponent intentions and plans on the basis of current situation estimation [3, 16, 29, 30]. There were published the studies on determining the malefactor’s plans during the intrusion detection [8, 10]. It is proposed to use the ideas of agent plans recognition on the basis of stochastic formal grammar recovery algorithms [11]. The important components in this research are the methods of reflexive processes theory [20], game theory [25] and control in conflict situations [6]. Authors used the methods of agent actions scenario specification based on the stochastic attributive formal grammars [11]. They correlated with colonies of cooperative distributed grammars and grammar models of multi-agent systems [17]. As teams are to adapt to reconfiguration, traffic changes and new types of defense and attacks on the basis of past experience it is important to take into account the present studies in the area of adaptation and self-learning [1, 13].

The rest of the paper is structured as follows. *Sections 1 and 2* outline suggested approach for modeling and simulation. *Section 3* describes the software environment developed for simulation. *Section 4* presents one of simulation scenarios fulfilled. *Conclusion* outlines the main results of the paper and future work directions.

1. Modeling and Simulation Approach

It is assumed the competing softbots gather information from different sources, operate with fuzzy (or probabilistic) knowledge, forecast the intentions and actions of opponent, estimate the possible risks and try to deceive each other, and react on opponent’s actions. The choice of behavior for each team depends on the chosen goal of functioning. The choice of every step of a team behavior is defined dynamically depending on the opposite team actions and the state of environment.

Each team acts in the conditions of limited information. Every team member might have different information about actions done by other team members. Therefore the model of behavior must be able to represent the incompleteness of information and the

possibility of accidental factors. The softbots are to foresee, what each softbot knows, what task has to be solved and to which one it must address its request if it is outside of its competence. The messages of one softbot are to be represented in such terms that are understandable by others.

The use of ontologies is the one of the most perspective approaches to structure the distributed knowledge. As for every application domain the information security ontology represents the partially normalized set of notions that are to be used by other softbots. Besides the relation of partial order the nodes of this structure have other relations peculiar to the application domain. The ontology defines the subset of notions, that various softbots use to solve tasks stated. Each softbot uses a fragment of application ontology. Each softbot specialization is represented by a subset of ontology nodes. Some of nodes can be shared by several softbots. Usually only one of them has the detailed description of a node. This softbot is the owner of the corresponding fragment of knowledge base. At the same time some part of ontological knowledge base is shared for all team. This part is the fragment that is to be the shared context.

The team of malefactors evolves with the aid of generation of new instances and types of attacks and attack scenarios. The defense team adapts by changing the security policy, forming new instances of defense methods and security profiles.

The softbots' counteraction model includes: (1) Ontology of application domain containing application notions and relations between them; (2) protocols of teamwork (for malefactors' and defense team); (3) Models of individual, group and team behavior; (4) Communication component for message exchange; (5) Models of environment (computer network), including topological and functional components.

The approach for teamwork proposed in the paper is based on the joint use of the elements of joint intentions theory, shared plans theory and hybrid approach. The teamwork is assumed to be organized due to the shared (group) plan of actions [19]. The structure of team is described in terms of group and individual roles hierarchy. The leaves of hierarchy correspond to the roles of individual softbots, the intermediate nodes – to the group roles. The specification of action plans hierarchy is made for every role. The following elements are defined for every plan: initial conditions, when the plan is offered for fulfillment; the conditions with which the plan stops being fulfilled; actions executed on the team level as the part of the shared plan. The joint activity is obviously expressed for the group plans. Softbots can create the "snapshots" of mental state of the whole team due to forming of joint intentions on the different abstract levels. The mechanisms of softbot interaction and coordination are based on the three groups of procedures [26, 19]: (1) the providing acts consistency; (2) softbots' functionality monitoring and recovery; and (3) communication selectivity support (to choose the most "useful" communication acts).

2. Attacks and Defense Softbots

The idea of *DDoS attack* consists in reaching the global goal – the denial of service of some resource – due to joint efforts of many components that are acting on attack side. In that way the initial goal is divided into more simple sub-goals. They are given to particular softbots. At the same time the goal on the top level stays shared between softbots. On the low level, the local goals are formed. Their achievement is targeted on solving the shared task. The softbots interact with each other to coordinate local solutions. This is necessary to reach the shared goal "denial of service".

Generally, the components of attack system are the programs which have the following features: autonomy; initial knowledge about itself, interacting entities and environment; knowledge (or hard-coded algorithm) that allows to process the external data from environment; the presence of a goal and a list of actions to reach this goal; the communication and interaction mechanisms (protocols) to reach the shared goal.

Analyzing the present methods of DDoS realization it is possible to determine at least two types of attack softbots: “Daemon” – it executes the attack directly; “Master” – it coordinates the actions of other system components. So the attack team is a two-level system. Masters act on the higher level directly fulfilling the malefactor’s tasks. They make decisions: when to start the attack, what target to attack, what is the attack intensity. Masters coordinate daemons’ actions by sending commands. Daemons act on lower level. After receiving the messages from masters, they start or finish sending the attack packets or change the attack intensity.

On the preliminary stage the master and daemons are deployed on available (compromised) hosts in the Internet. Then the attack team is established: daemons send to master the messages saying they are alive and ready to work. Master stores the information about team members and their state. The malefactor sets the common goal of team – to perform DDoS attack. Master receives attack parameters. Its goal is to distribute these parameters among all available daemons. Then daemons act. Their local goal is to execute the master command. To fulfill attack they send the attack packets to the given host. After this it is believed that the goal is reached. Master asks daemons periodically to find out that they are alive and ready to work. Receiving the messages from daemons the master manages the given rate of attack. If there is no any message from one of the daemons the master makes the decision to change the attack parameters. For example, it can send to some or all daemons the commands to change the attack rate. Daemons can execute the attack in various modes. This feature affects on the potentialities of defense team. Daemons can use different types of attacks, send attack packets with various rates, spoof source IP address and do it with various rates. Malefactor can stop the attack by sending to master the command “stop the attack”. Then master distributes this command among all daemons, and they stop the attack.

The main task of *defense systems against DDoS* is to accurately detect attacks and quickly respond to them [31]. It is equally important to recognize the legitimate traffic that shares the attack signature and deliver it reliably to the victim [21]. *Traditional defense* include detection and reaction mechanisms. Different network characteristics are used for detection of malicious actions (for example, source IP address [24], traffic volume [9], and packet content [23]). To detect abnormal network characteristics, many methods can be applied (for instance, statistical, cumulative sum, pattern matching, etc). As a rule, the reaction mechanisms include filtering, congestion control and traceback. But, as a result of several reasons (detection of DDoS attack is most accurate close to the victim, separation of legitimate is most successful close to the sources, etc.), adequate victim protection to constrain attack traffic can only be achieved by *cooperation of different distributed components* [21]. So, the DDoS problem requires a distributed cooperative solution [2, 4, 23, 18, 32, 31, 21, etc.].

The analysis of present DDoS defense systems shows the following their features: The defense systems are built of basic components which have some local meaning but serve together for common shared goal; The number and functionality of defense system components depend on the place of their deployment; As a rule, the defense systems have a hierarchical structure, where different levels serve for particular sub-tasks of the complex defense goal. The general approach to the DDoS defense is the

following. The information about normal traffic is collected from different network sensors. Then the analyzer-component compares in real-time the current traffic with the normal traffic. The system tries to trace back the source of anomalies (due to “traceback” mechanisms) and generates the recommendations how to cut off them or how to lower the quantity of these anomalies. Depending on security administrator’s choice, the system applies some countermeasure. We set the following defense softbot classes: “Sensor” – for initial information processing; “Sampler” – the network data collector that forms the traffic model; “Detector” – for attack detection; “Filter” – for attack traffic filtering; “Investigator” – for attack investigation. *Sensor* processes information about network packets and collects statistic data on traffic for defended host. Sensor determines the size of overall traffic (*BPS* – *bit per seconds*) and the addresses of n hosts that make the greatest traffic (in developed prototype – all hosts). Its local goal is to give these parameters to detector every k seconds. *Samplers* are deployed in the defended subnet to collect the data on its normal functioning. Using this data they can detect anomalies. The examples of methods which can be realized by sampler are Hop counts Filtering (HCF) [15], Source IP address monitoring (SIPM) [28], Bit per Second (BPS), etc. *Detector* local goal is to make the decision if the attack happens. It sends its decision and N addresses to filter and to investigator. *Filter* local goal is to filter the traffic on the basis of data from detector. If it was determined that the network is under attack, filter begins to filter the packets from the given hosts. The goal of *investigator* is to identify and defeat the attack softbots. When investigator receives the message from detector it examines the given addresses on the presence of attack softbots and tries to defeat identified softbots.

3. Simulation Environment

The simulation environment architecture consists of the following components (Fig.1): OMNeT++ Framework, INET Framework, Multi-agent & DDoS Framework. Agent-based simulation is implemented in Multi-agent Framework that uses the library of attack and defense mechanisms called DoS Framework. INET Framework is used to simulate the IP nodes. It is an OMNeT++ model itself.

OMNeT++ Framework [27] is a discrete event simulator. Its primary application area is the simulation of computer networks and other distributed systems. Simulation models are composed of hierarchically nested modules that interact due to message passing (Fig.1, OMNeT++ Framework: simulation model & component library). Module functionality is programmed using C++, while the model structure is defined by the special topology description language. INET Framework and Multi-agent DDoS Framework are the OMNeT++ models. The exchange of messages between modules happens due to channels (modules are connected with them by the gates) or directly by gates. A gate can be incoming or outgoing to receive or to send messages accordingly. Channel is characterized by propagation delay, bit error rate and transmission data rate. OMNeT++ INET Framework is the OMNeT++ modular simulation suite with a realistic simulation of Internet nodes and protocols. The highest IP simulation abstraction level is the network itself, consists of IP nodes. IP node can represent router or host. IP node in INET Framework corresponds to the computer representation of Internet Protocol (Fig.1, INET Framework). Multi-agent & DDoS Framework is the INET Framework modular suite aimed to simulate the DDoS attack and defense mechanisms on the basis of team counteraction (Fig.1, Multi-agent DDoS Framework).

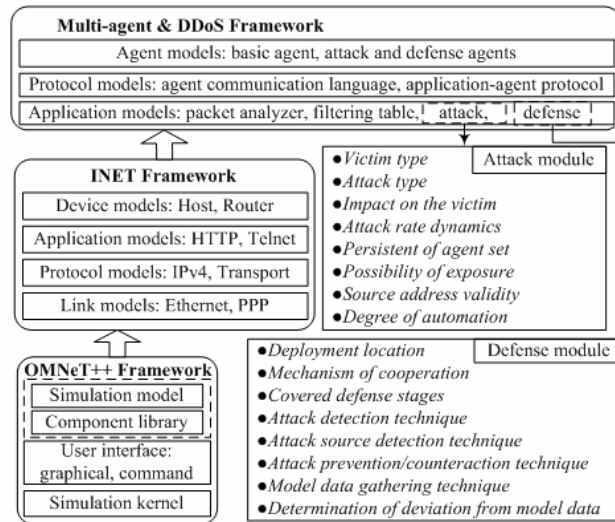


Figure 1. Simulation environment architecture

DDoS Framework suite consists of DDoS attack and defense modules (Fig.1, Attack module, Defense module) and the modules that expand IP node from INET: the filtering table and the packet analyzer. Attack and defense modules are the applications and are deployed on the network layer of IP node. To set the DDoS attack conditions it is necessary to define the corresponding *input parameters*, including victim type (host), attack rate dynamics (function of attack packets sending rate), spoofing technique (no spoofing, random, subnet), etc. Also one need to set up the defense parameters, including deployment location (defended, intermediate, source subnet), detection technique, model data gathering technique and its parameters (time interval and time shift of data collection), etc. The examples of *output parameters* used to estimate the defense are as follows: Time of attack detection; Time of attack reaction (time from detection to counteraction); Percent of false positives; Percent of false negatives; Percent of normal traffic filtration; Computational complexity, etc.

Agent Framework consists of modules representing softbots implemented as applications. There were used the elements of abstract FIPA architecture [7] during softbot modules design and implementation. Agent communication language is implemented for softbot interactions. The message passing happens above the TCP protocol (transport layer). Softbot can control the other modules (including DDoS Framework modules) due to messages. Softbots are deployed on the hosts in the simulation environment. Their installation is fulfilled by connecting to the modules serving transport and network layers of protocol stack simulated in INET Framework.

The example of multi-window user interface of the simulation environment is depicted in Fig.2. The window for simulation management (at the bottom of Fig.2, at right) allows looking through and changing simulation parameters. It is important that you can see the events which are very valuable for understanding attack and defense mechanisms on time scale. Corresponding status windows (on top of Fig.2, in the middle and at left) show the current status of teams. It is possible to open different windows which characterize functioning of particular hosts, protocols and softbots, for example, at the bottom left of Fig.2, the window of one of the hosts is displayed.

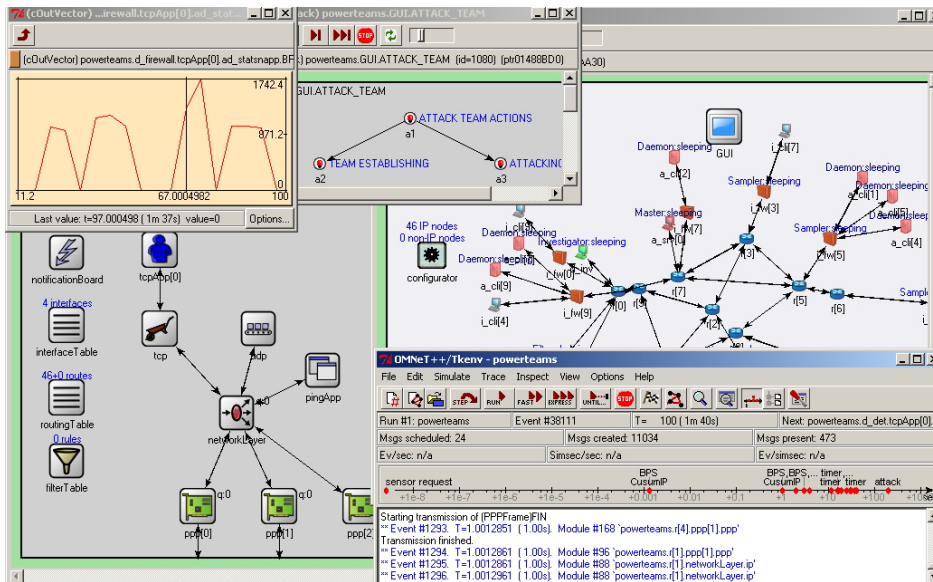



Figure 2. Common representation of simulation environment

At the basic window of visualization (Fig.2, at upper right), a simulated computer network is displayed. The network represents a set of hosts and channels. Hosts can fulfill different functionality depending on their parameters or a set of internal modules. The routers are labeled with the sign “”. Attack softbots are deployed on the hosts marked with red color. Defense softbots are located on the hosts marked with green color. Above the colored hosts there are the strings that indicate the corresponding state of deployed softbots. The other hosts are the standard hosts that generate generic traffic.

Each network for simulation consists of three sub-networks: (1) the subnet of defense where the defense team is deployed; (2) the intermediate subnet where the standard hosts are deployed. They produce the generic traffic in the network including the traffic to defended host; (3) the subnet of attack where the attack team is deployed.

4. Simulation Example

Learning mode. The main task of learning mode is to create the model of generic traffic. The clients send the requests to the server and it replies. At this time sampler analyses requests and uses them to form the models and parameters. Fig.3 depicts the change of new addresses amount for sampler during first 300 seconds of learning. Fig.4 represents the graph of change of maximum BPS (for interval 10 seconds and shift 3 seconds) after 300 seconds from the beginning of learning.

Decision making and acting. Simulation scenario is realized on the same configuration as was used during learning. The only difference – the attack team is engaged. Attack initial parameters are as follows: target of attack is server d_srv; intensity of attack – 5 packets per sec); no IP spoofing is used.

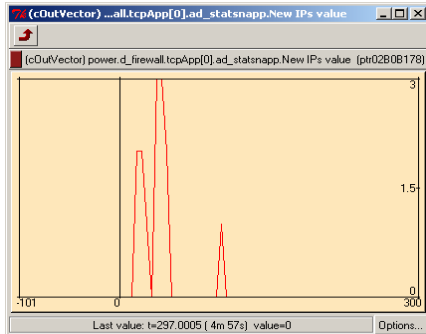


Figure 3. Change of new IP addresses amount

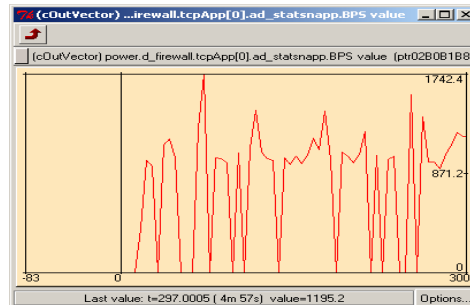


Figure 4. Change of BPS parameter

Fig.5 represents the graphs of channel throughput (bits/s to sec) on the entrance to the defended network before (red) and after (blue) filter. After simulation start the clients begin to send requests to the server and it replies. This is the way the generation of generic network traffic takes place (Fig.5, interval 0 – 300 sec). After establishing the defense team begins to function. Sampler collects traffic data and compares it with the model data that it acquired during learning mode. The addresses that are the source of anomalies are sent to detector. Detector makes the decision about the attack and sends to filter and investigator the addresses of suspicious hosts.

After 300 sec from simulation start the attack team begins attack actions. When daemons receive the attack command they begin to send the attack packets (Fig.5, timestamp 300 sec). After a while, sampler determines the suspicious hosts with the use of BPS method. The BPS parameter of these hosts exceeds normal. Detector receives the addresses of these hosts from sampler and sends them to filter and investigator. Filter sets the filtering rules and the packets from the given hosts begin being dropped (Fig.5, timestamps 400 – 600 seconds, blue graph).

Investigator tries to inspect the given hosts and to defeat the attack softbots deployed there. It succeeds in defeating of 4 daemons. However the other daemons continue the attack (Fig.5, after 400 seconds, red graph). Master makes the decision to redistribute the intensity of attack to keep the overall intensity on the given level. Also it decides to change the method of IP spoofing to complicate the detection and defeating of attack softbots by defense team. Master sends to alive daemons the command: target – d_srv, target port – 2001, intensity – $5/(10-4)=0.83$, IP spoofing method – “random”. When daemons receive the command they continue to send the attack packets having applied the new parameters (Fig.5, timestamp 600 sec).

Detector sees that the input channel throughput has noticeably lowered since the traffic from attack team has raised (Fig.5, after 600 sec). Detector does not receive the anomaly report from sampler though. This is because the method BPS used by sampler does not work fine when attacker changes the sender address in every packet. That is the reason detector fails to confront some address with the big traffic. Therefore detector decides to apply another DDoS defense method – SIPM. The large amount of new IP addresses for sampler will lead to attack detection and dropping of malicious packets. This method however does not allow tracing the source of attack and investigator will fail to defeat attack softbots. But the attack packets will be filtered and the traffic in the subnet of defended host will return to normal state.

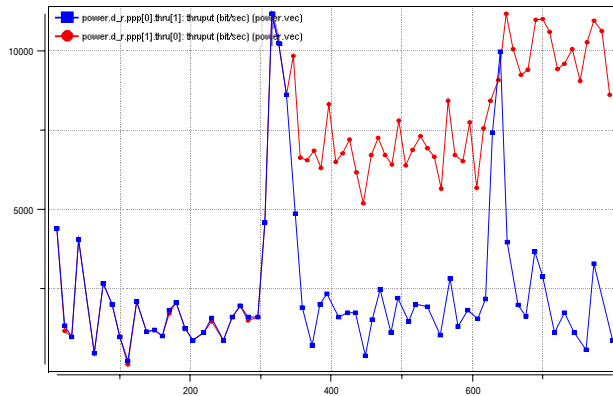


Figure 5. Graphs of channel throughput

5. Conclusion

The main results of the paper consist in developing a new agent-based approach for comprehensive investigation of defense mechanisms against distributed coordinated attacks in the Internet and implementing the software environment (written in C++ and OMNeT++) intended for simulation of DDoS attacks. We tried to model and simulate counteraction between malefactors and defense systems in the Internet as competition between teams of softbots. The main attention was drawn to the application of packet-based imitation of softbots' interaction which provides the acceptable fidelity and scalability of computer attack and defense mechanisms representation. The goal of the paper is not to present an investigation of new defense methods, but to show the possibilities of the approach suggested and the knowledge-based simulation tool developed. One of the features of this tool is the possibility to insert new attack and defense methods and investigate them. So the approach suggested and the environment implemented can be extended for investigation of other classes of attacks.

The approach was examined on the example of "Distributed Denial of Service" attacks and defense simulation. We considered different phases of operations of antagonistic softbots' teams – learning, decision making and counteracting, including adaptation of one team to the actions of opposite team. Various experiments with this environment have been fulfilled. These experiments include the investigation of attack scenarios and protection mechanisms for the networks with different structures and security policies. One of the scenarios was demonstrated in the paper.

Future work is connected with developing an expressive formal framework for specification of softbots' competition and collaboration in the Internet, building a more powerful simulation environment, investigating new defense mechanisms, and conducting experiments to both evaluate new large-scale network defense mechanisms and analyze the efficiency and effectiveness of different security policies which can be implemented for practical security solutions in the Internet.

This research is being supported by grant of Russian Foundation of Basic Research (№ 04-01-00167), grant of the Department for Informational Technologies and Computation Systems of the Russian Academy of Sciences (contract №3.2/03) and partly funded by the EC as part of the POSITIF project (contract IST-2002-002314).

References

- [1] T.Back, D.B.Fogel, Z.Michalewicz, *Evolutionary computation. Vol. 1. Basic algorithms and operators*, Institute of Physics Publishing. (2000).
- [2] R.Canonico, D.Cotroneo, L.Peluso, S.P.Romano, G.Ventre, Programming routers to improve network security. *Proceedings of the OPENSIG Workshop*. (2001).
- [3] E.Charniak, R.P.Goldman, A Bayesian Model of Plan recognition. *Artificial Intelligence*, V.64, N 1. (1993).
- [4] S.Chen, Q.Song, Perimeter-Based Defense against High Bandwidth DDoS Attacks. *IEEE Transactions on Parallel and Distributed Systems*, Vol.16, No.7. (2005).
- [5] P.R.Cohen, H.J.Levesque, Teamwork. *Nous*, Vol.25, No.4. (1991).
- [6] V.V.Druzhinin, D.S.Kontorov, M.D.Kontorov, *Introduction into conflict theory*. Moscow, Radio i svyas' (1989) (in Russian).
- [7] FIPA. <http://www.fipa.org>
- [8] C.W.Geib, R.P.Goldman, Plan recognition in intrusion detection systems. *DARPA Information Survivability Conference and Exposition*, DARPA and the IEEE Computer Society. (2001).
- [9] T.M.Gil, M.Poletto, MULTOPS: a data-structure for bandwidth attack detection. *Proceedings of 10th Usenix Security Symposium*. (2001).
- [10] R.P.Goldman, C.W.Geib, C.A.Miller, A New Model of Plan Recognition. *Proceedings of the 1999 Conference on Uncertainty in Artificial Intelligence*. (1999).
- [11] V.Gorodetski, I. Kotenko, Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool. *Lecture Notes in Computer Science*, V.2516. (2002).
- [12] B.Grosz, S.Kraus, Collaborative plans for complex group actions. *Artificial Intelligence*, Vol.86. (1996).
- [13] D.Gu, E.Yang, Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey, *Technical Report of the Department of Computer Science, University of Essex, CSM-404*. (2004).
- [14] N.R.Jennings, Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, Vol.75, No.2. (1995).
- [15] C.Jin, H.Wang, K.G.Shin, Hop-count filtering: An effective defense against spoofed DDoS traffic. *Proceedings of the 10th ACM Conference on Computer and Communications Security*. (2003).
- [16] H. Kautz, J.F.Allen, Generalized plan recognition. *Proceedings of the Fifth National Conference on Artificial Intelligence*. (1986).
- [17] J.Kelemen, Colonies: grammars of reactive systems. *Proceedings of AICRS'97*. (1997).
- [18] A.D.Keromytis, V.Misra, D.Rubenstein, SOS: An architecture for mitigating DDoS attacks. *Journal on Selected Areas in Communications*, Vol. 21. (2003).
- [19] I.Kotenko, A.Ulanov, Multiagent modeling and simulation of agents' competition for network resources availability. *Second International Workshop on Safety and Security in Multiagent Systems, Utrecht, The Netherlands*. (2005).
- [20] V.A.Lefevre, *Reflexion*. Moscow, "Kognito-Center" (2003) (in Russian).
- [21] J.Mirkovic, M.Robinson, P.Reiher, G.Oikonomou, Distributed Defense Against DDOS Attacks. *University of Delaware. CIS Department. Technical Report CIS-TR-2005-02*. (2005).
- [22] J.Mirkovic, S.Dietrich, D.Dittrich, P.Reiher, Internet Denial of Service: Attack and Defense Mechanisms. Prentice Hall PTR. (2004).
- [23] C.Papadopoulos, R.Lindell, I.Mehringier, A.Hussain, R.Govindan, Cossack: Coordinated suppression of simultaneous attacks. *Proceedings of DISCEX III*. (2003).
- [24] T.Peng, L.Christopher, R.Kotagiri, Protection from Distributed Denial of Service Attack Using History-based IP Filtering. *IEEE International Conference on Communications*. (2003).
- [25] A.A. Stogniy, A.I. Kondrat'ev *Game theory information modeling in decision making systems*. Kiev: Naukova dumka. (1986) (in Russian).
- [26] M.Tambe, Towards flexible teamwork. *Journal of AI Research*, Vol.7. (1997).
- [27] OMNeT++ homepage. <http://www.omnetpp.org/>
- [28] T. Peng, C. Leckie, and R. Kotagiri, Proactively Detecting DDoS Attack Using Source IP Address Monitoring, *Networking 2004*, Athens, Greece. (2004).
- [29] M.Vilain, Getting Serious about Parsing Plans: A Grammatical Analysis of Plan Recognition. *Proceedings of the Eighth National Conference on Artificial Intelligence*, Cambridge, MA. (1990).
- [30] M.P.Wellman, D.V.Pynadath, *Plan Recognition under Uncertainty*, Unpublished web page. (1997).
- [31] Y.Xiang, W.Zhou, An Active Distributed Defense System to Protect Web Applications from DDoS Attacks. *The Sixth International Conference on Information Integration and Web Based Application & Services*. (2004).
- [32] D.Xuan, R.Bettati, W.Zhao, A gateway-based defense system for distributed dos attacks in high-speed networks. *IEEE Transactions on Systems, Man, and Cybernetics*. (2002).