IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
6-8 September 2007, Dortmund, Germany

Multi-agent Modelling and Simulation of Cyber-Attacks and Cyber-Defense for Homeland Security

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

Abstract *– The paper considers the approach to investigation of distributed cooperative cyber-defense mechanisms against network attacks. The approach is based on the agent-based simulation of cyber-attacks and cyber-protection mechanisms which combines discrete-event simulation, multi-agent approach and packet-level simulation of network protocols. The various methods of counteraction against cyber-attacks are explored by representing attack and defense components as agent teams using the developed software simulation environment. The teams of defense agents are able to cooperate as the defense system components of different organizations and Internet service providers (ISPs). The paper represents the common framework and implementation peculiarities of the simulation environment as well as the experiments aimed on the investigation of distributed network attacks and defense mechanisms.*

Keywords *– cyber-attacks, cyber-defense, agents, simulation*

I. Introduction

Today we are in growing dependence from information and telecommunication technologies. The further use of them becomes impossible without appropriate protection mechanisms and effective homeland security solutions. The important problem in homeland security which solution is urgently needed is *the investigation of counteraction between malefactors and defense systems in computer networks, including the Internet, and the creation of prospective intelligent cyber-defense systems*.

The design and implementation of effective cyber-defense intelligent cyber-defense system is a very complicated problem. According to our view the prospective network cyber-defense systems has to be *fully integrated and multi-echeloned* ones. To effectively detect the computer attacks or unauthorized operations and to flexibly react on them, it is needed to carry out the continuous control of network functioning, analyze possible risks, collect knowledge about counteraction, detection and reaction methods and use them for defense reinforcement. Besides, the effective cyber-defense should include the mechanisms of attack prevention, detection, source tracing and protection as well as can only be achieved by the *cooperation of different distributed components* [17, 18]. For example, detection of Distributed Denial of Service (DDoS) flooding attack [3, 12, 16, 23, 24, 27] is most accurate close to the victim, but separation of legitimate is most successful close to the sources, therefore the security sub-systems (or teams) have to be located at different network places and tightly cooperate (see in Fig. 1 the defense teams 1, 2 and N).

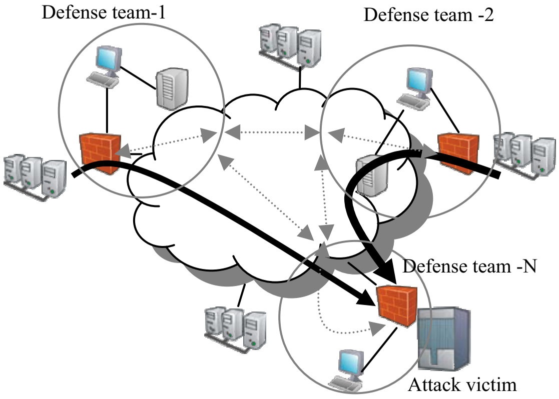


Fig. 1. Cooperation of agent teams in the Internet.

The cyber-defense systems have to be *adaptive and evolve dynamically* with the change of network conditions. To implement these possibilities in prospective cyber-defense system one must implement the dynamic behavior, autonomy and adaptation of particular components, the use of methods based on negotiations and cooperation that lie in the basis of multi-agent systems and (or) autonomic computing.

Furthermore, the prospective cyber-defense system has to provide at least *three levels of cyber-security*. *First level* contains "traditional" static cyber-defense mechanisms implementing identification and authentication, cryptographic protection, access control, auditing, network filtering, etc. *Second level* includes proactive cyber-defense mechanisms that provide information collection, security assessment, network state monitoring, attack detection and counteraction, malefactor deception, etc. *Third level* corresponds to cyber-defense management that fulfills the integral evaluation of network state, the choice of adequate or optimal defense mechanisms and their adaptation. This level is built on top of various non-adaptive security mechanisms, which makes it applicable for a wide range of cyber defenses.

The paper considers *the approach to multi-agent simulation of cyber-attacks and distributed cooperative multi-level cyber-defense for the exploration of prospective intelligent cyber-defense systems*. We analyze various methods of counteraction against cyber-attacks by representing attack and defense components as agent teams using the simulation environment developed. Various teams of defense agents are able to cooperate as the defense system components of different organizations and Internet service providers (ISPs). The rest of the paper is structured as follows. *Section 2* outlines the common multi-agent modeling and simulation framework suggested. *Section 3* describes the implementation peculiarities of the simulation environment developed. *Section 3 presents* demonstrates the examples of experiments provided with the simulation environment. Conclusion surveys the main results of the paper.

II. Multi-Agent Simulation Framework

The multi-agent approach to simulation supposes that the cyber-counteraction is represented as the interaction of different teams of software agents [17, 18]. The aggregated system behavior becomes apparent by means of local interactions of particular agents in dynamic environment that is defined by the model of computer network.

There are at least three different classes of agent teams [19]: teams of agents-malefactors, teams of defense agents, teams of agents-users. Agents of different teams can be in indifference ratio, cooperate or compete up till explicit counteraction. Agents are supposed to collect information from various sources, operate incomplete knowledge, forecast the intentions and actions of other agents, try to deceive the agents of competing teams, react to actions of other agents. Every team member might have different information about actions done by other team members. Therefore, the model of agent behavior must be able to represent the incompleteness of information and the possibility of accidental factors. Besides, the agent behavior depends on information that the team has and on its distribution on the set of particular agents. The models of agent functioning are to foresee, what each agent knows, what task has to be solved and to which agent it must address its request to receive such information if it is outside of its competence.

The *general conceptual model of cybernetic agents' counteraction and cooperation* includes (Fig. 2) [8, 19]: the ontology of application containing application notions and relations between them; the protocols of teamwork for the agents of different teams; the models of scenario behavior of agents for team, group and individual levels; the libraries of agent basic functions; the communication platform and components for agent message exchange; the models of the computer network as the communication environment. As for every application domain the information security ontology represents the partially normalized set of notions that are to be used by other agents. The given ontology defines the subset of notions that various agents use for cooperative solving of stated tasks. Each agent uses a certain part of application domain ontology. Each agent specialization is represented by a subset of ontology nodes.

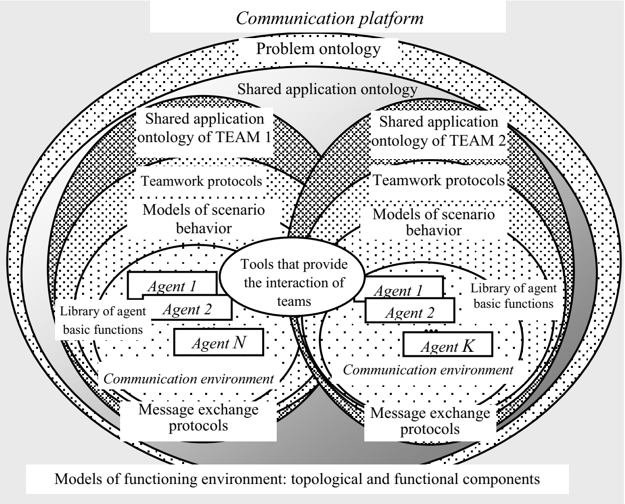

Communication platform
Problem ontology
Shared application ontology
Shared application ontology of TEAM 1
Shared application ontology of TEAM 2
Teamwork protocols
Teamwork protocols
Models of scenario behavior
Models of scenario behavior
Tools that provide the interaction of teams
Agent 1
Agent 2
Agent N
Library of agent basic functions
Agent 1
Agent 2
Agent K
Library of agent basic functions
Communication environment
Communication environment
Message exchange protocols
Message exchange protocols
Models of functioning environment: topological and functional components

Fig. 2. Abstract model of team interaction.

Attack agents are subdivided at least into two classes: "daemons" and "masters". To simulate distributed cooperative defense, the security agents belong to the following classes: information processing ("samplers"); attack detection ("detectors"); filtering and balancing ("filters"); traceback and investigation ("investigators").

It is supposed that agents are to be able to realize the mechanisms of self-adaptation and evolution. The team of agents-malefactors evolves with the aid of generating new attacks and attack scenarios to overcome the defense. The team of defense agents adapts by changing the security policy and defense methods and profiles.

The main basis for the research is the agent teamwork approaches: joint intentions theory [4], shared plans theory [9] and the hybrid approaches [31]. Another fundamental component of the research is represented by the studies on reasoning systems about opponent intentions and plans [2, 7, 32, 34]. The important components in this research are the methods of reflexive processes theory [21], game theory and control in conflict situations [5]. The teams of malefactors and defense agents are to adapt to network reconfiguration, traffic changes and new types of defense and attacks on the basis of past experience. Therefore it is important to take into account the present studies in the area of adaptation [30, 20], agent learning [1, 6, 10, 35], autonomic computing [11, 15, 33], and combining artificial immune systems with different computational intelligence methods, such as fuzzy systems, neural networks, etc. [13, 25].

III. Simulation Environment

A multi-level software environment was supposed to be developed to implement the proposed approach. It differs from the known tools for agent-oriented simulation, as the basis for simulation the tools should be used which provide adequate simulation of network processes. The spectrum of possible approaches to modeling and simulation are differentiated from analytical to scaled-down and full-scale (see Fig. 3) [29].

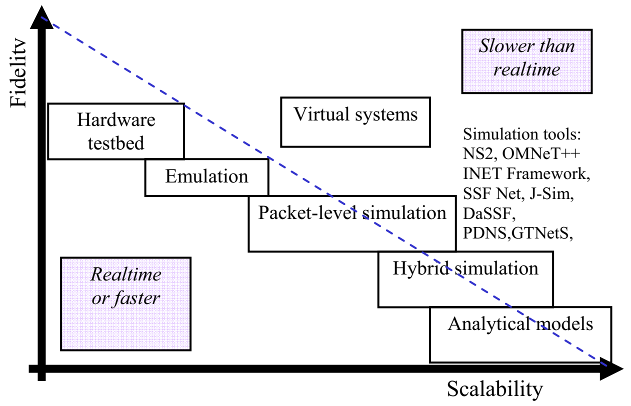


Fig. 3. Variety of used models.

All methods are considered in two basic dimentions: scalability and fidelity. The scalability is defined as number of network host (client hosts and routers) which can be simulated using the given method. The fidelity is defined as a degree of network and hosts destabilization used.

We have chosen the *packet-based approach* as it provides acceptable scalability and fidelity. In Fig. 3, various program simulators which can be used for simulation are depicted: NS2, OMNeT++ INET Framework, SSF Net, J-Sim, etc. To choose the necessary tool we fulfilled the detailed analysis of these simulation environments [18] and OMNeT++ INET Framework was chosen [26].

The simulation environment architecture suggested includes the following components (Fig. 4): Simulation Framework, Internet Simulation Framework, Multi-agent Simulation Framework, Subject Domain Library.

Simulation framework is a discrete event simulator. Other components are expansions or models for Simulation Framework.

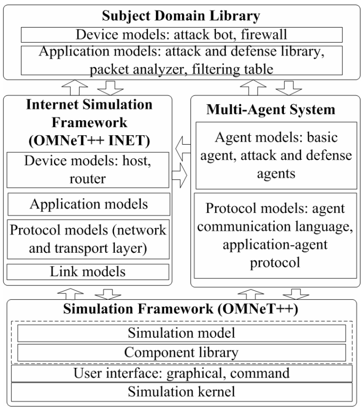

Subject Domain Library
Device models: attack bot, firewall
Application models: attack and defense library, packet analyzer, filtering table
Internet Simulation Framework (OMNeT++ INET)
Device models: host, router
Application models
Protocol models (network and transport layer)
Link models
Multi-Agent System
Agent models: basic agent, attack and defense agents
Protocol models: agent communication language, application-agent protocol
Simulation Framework (OMNeT++)
Simulation model
Component library
User interface: graphical, command
Simulation kernel

Fig. 4. Simulation environment architecture.

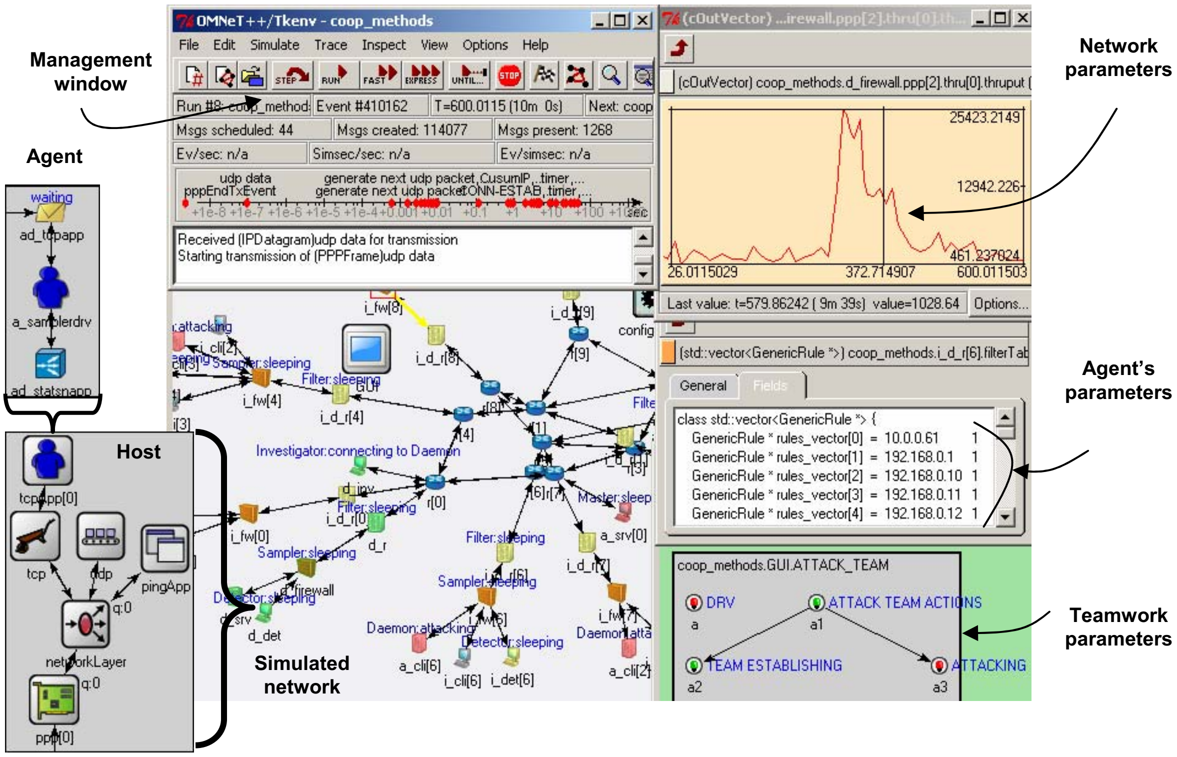
Internet Simulation Framework is a modular simulation suite with a realistic simulation of Internet nodes and protocols. The highest IP simulation abstraction level is the network itself, consisting of IP nodes. IP node corresponds to the computer representation of Internet Protocol. IP node can represent router or host. IP node in Internet Simulation Framework corresponds to the computer representation of Internet Protocol. The modules of IP node are organized as operating system process IP datagram. The module that is responsible for the network layer (implementing IP processing) and the “network interface” modules are mandatory. In addition one can plug the modules that implement higher layer protocols.

Multi-agent Simulation Framework allows realizing agent-based simulation. It consists of modules representing the intelligent agents implemented as applications. There were used the elements of abstract FIPA architecture during agent modules design and implementation. Agent communication language is implemented for the agent interactions. The message transmission occurs above the TCP protocol (transport layer) implemented in Internet Simulation Framework. Agent directory is mandatory only for agent that coordinates other agents in its team. Agent can control the other modules due to messages.

Subject Domain Library is the library used for imitation of processes from subject domain and containing modules that extend functionality of IP-host: filtering table and packet analyzer.

This architecture was implemented for multi-agent simulation DDoS attack and defense mechanisms with the use of OMNeT++ INET Framework and software models developed in C++. Agent models implemented in Multi-agent Simulation Framework are represented with generic agent, attack and defense agents. Subject Domain Library contains various models of hosts, e.g. attacking host, firewall etc., and also the application models (attack and defense mechanisms, packet analyzer, filtering table).

Fig. 5 shows the *multi-window user interface of the simulation environment*. The management window has the time axis with the system events: opening or closing the TCP connection, attack signals, defense acts, etc. The simulated network window depicts the network hosts and channels. Hosts can fulfill different functionality depending on their parameters or a set of internal modules. Internal modules are responsible for functioning of protocols and applications at various levels of OSI model. Hosts are connected by channels which parameters can be changed. Applications (including agents) are established on hosts. Applications are connected to corresponding modules of protocols. The structure of generic host is depicted on the bottom left. The deployed agent is represented as the blue symbol of human in the frame. The environment allows to examine the different information describing the simulation functioning. For example, the diagram that shows the change in network parameters is depicted at the top right. The networks used



OMNeT++/Tkenv - coop_methods

Run #8: coop_method: Event #410162 T=600.0115 (10m 0s) Next: coop

Msgs scheduled: 44 Msgs created: 114077 Msgs present: 1268

Ev/sec: n/a Simsec/sec: n/a Ev/simsec: n/a

```
Received (IPDatagram)udp data for transmission
Starting transmission of (PPPFrame)udp data
```

(cOutVector) ...irewall.ppp[2].thru[0].th...

(cOutVector) coop_methods.d_firewall.ppp[2].thru[0].thruput (

Last value: t=579.86242 (9m 39s) value=1028.64

(std::vector<GenericRule *>) coop_methods.i_d_r[6].filterTab

```
class std::vector<GenericRule *> {
   GenericRule * rules_vector[0] = 10.0.0.61     1
   GenericRule * rules_vector[1] = 192.168.0.1   1
   GenericRule * rules_vector[2] = 192.168.0.10  1
   GenericRule * rules_vector[3] = 192.168.0.11  1
   GenericRule * rules_vector[4] = 192.168.0.12  1
```

coop_methods.GUI.ATTACK_TEAM

Fig. 5. Multi-window user interface.

for simulation consist of various subnets that are, for example, the regions of responsibility of various ISPs. For example, one can mark out the defense subnet where the attack victim is located, the intermediate subnets where the standard hosts generate generic network traffic, and attack subnets where the attack agents are located. The networks are built with the methods of generating topologies that are close to the real Internet [22].

IV. EXPERIMENTS

The developed simulation environment allows carrying various experiments aimed to investigate attacks and prospective defense strategies. One can vary network topology and configuration, structure and configuration of attack and defense teams, attack and defense mechanisms, team cooperation parameters etc. The evaluations of various effectiveness parameters of defense mechanisms are done on the basis of experiments results. The analysis of applying conditions is also fulfilled for these parameters.

The attack parameters used in the experiments are as follows: Victim type – host (server that provides some service); Attack type – brute-force; Impact on the victim – disruptive; Attack rate dynamics – constant, variable; Agents' set permanency – constant, variable; Possibility of exposure – discoverable filterable attack; Source addresses validity – valid (real), spoofed: random, subnet; Degree of automation – semi-automatic with direct communication.

In the experiments we have used three defense methods: Hop counts Filtering (HCF) [14], Source IP address monitoring (SIPM) [28] и Bit Per Second (BPS). *HCF* consists in building the tables of subnets and amount of hops till them in the learning mode. Attack is found out on the basis of amount of hops differing from received in learning mode. *SIPM* uses the assumption that during attack a lot of new IP addresses appear. *BPS* allows detecting the attacker due to exceeding the normal traffic threshold. The other defense parameters are as follows: Deployment location – intermediate, defended subnets; Covered defense stages – attack prevention, attack detection, attack source detection, attack counteraction; Attack source detection technique – can detect when source address is not spoofed; Attack prevention technique – packet filtering; Technique for gathering of model data – learning; Determination of deviation from model data: thresholds (HCF, BPS), determination of fluctuation in probabilistic traffic parameter (SIPM).

Let us consider two examples of experiments fulfilled where we analyzed different modes of DDoS attacks and defense:

- Experiment 1: Investigation of simple adaptation of attack and defense teams.
- Experiment 2: Investigation of different cooperation modes between defense teams.

The fragment of the network which was used in the *first experiment* is depicted in Fig. 6.

The fragment of decision making and acting sequence is as follows (Fig 7):

- Normal work of users (interval 0 – 300 seconds).
- Defense team: formation of the team; the team start using BPS method.

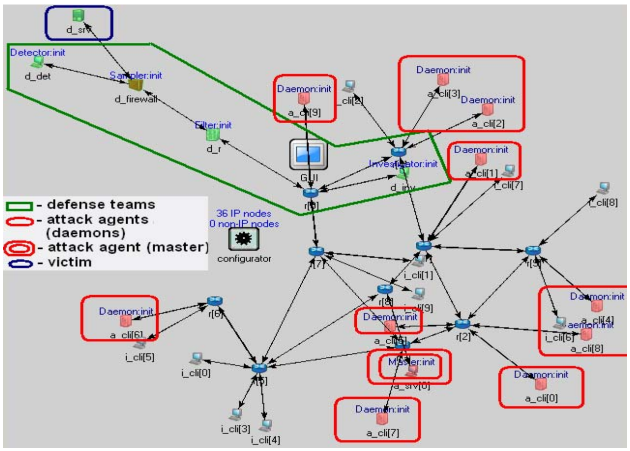


Fig. 6. Experiment 1: the Internet fragment and agent teams.

- Attack team: formation of the team; after 300 seconds the team begins the attack actions (intensity of attack for every daemon - 0.5, no IP spoofing).
- Defense team: data processing, attack detecting (using BPS) and reacting (interval 300 – 350 seconds); blocking the attack, destroying some attack agents (interval 300 – 600 seconds).
- Attack team: after 600 seconds the automatic adaptation is fulfilled (redistributing the intensity of attack (0.83), changing the method of IP spoofing (Random)).
- Defense team: data processing, failing to detect the attack (using BPS method) – Detector sees that the input channel throughput has noticeably lowered, but does not receive any anomaly report from sampler because BPS does not work.
- Defense team: Changing defense method on SIPM (automatic adaptation); Data processing, attack detecting (using SIPM method) and reacting – (interval 600 – 700 seconds).

The fragment of the network which was used in the *second experiment* is depicted in Fig. 8.

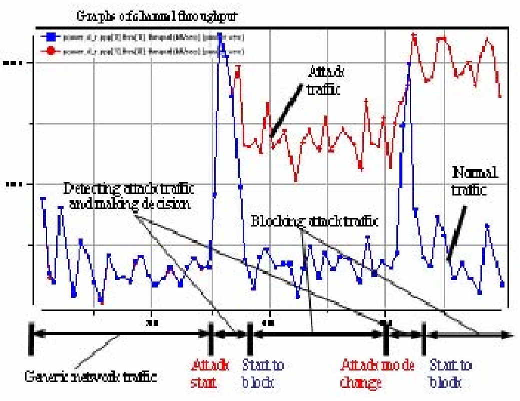


Fig. 7. Scheme of teams' acting.

We investigated *the models of cooperation between distributed defense teams*: (1) *filter-level cooperation:* the team whose network is under attack can apply filtering rules on the filters of other teams; (2) *sampler-level cooperation:* the team whose network is under attack can get the traffic information from the samplers of other teams; (3) *"poor" cooperation*: the teams can get the traffic information from the samplers of some other teams and apply filtering rules on the filters of some other teams (each team knows a subset of other teams depending on the cooperation degree); (4) *"full" cooperation*: the team whose network is under attack can get the traffic information from all samplers of other teams and apply filtering rules on all filters of other teams.

The volume of input traffic before and after the filter of the team which network is under attack when the BPS method is used is depicted in Fig. 9.

The other *effectiveness and efficiency parameters of different defense mechanisms* which were investigated are as follows: rate of dropped legitimate traffic (false positive rate); rate of admitted attack traffic (false positive rate); attack reaction time.

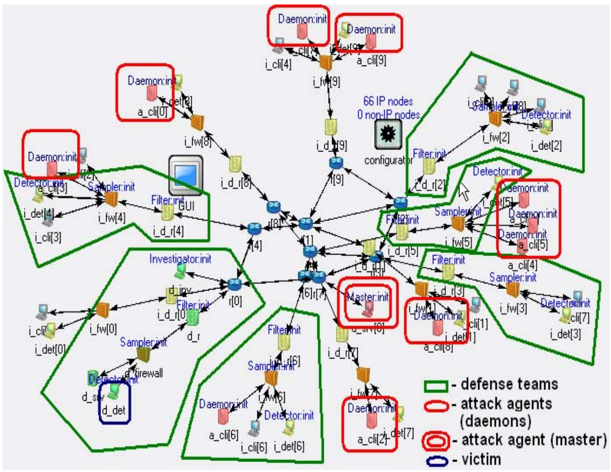
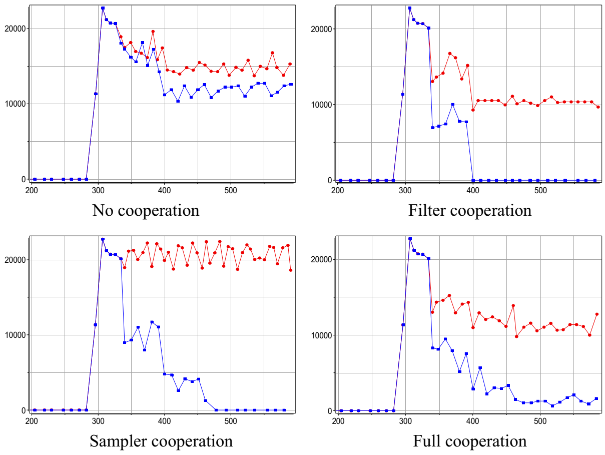


Fig. 8. Experiment 2: the Internet fragment and agent teams.



No cooperation

Filter cooperation

Sampler cooperation

Full cooperation

Fig. 9. Volume of input traffic before and after the filter.

These parameters were investigated in dependence on the following *input parameters*: network configuration; attack intensity; IP address spoofing technique used in attack; internal parameters of defense mechanisms and their combinations; quantity and distribution of defense teams, etc.

V. Conclusions

This paper considered the approach to investigation of distributed cooperative cyber-defense mechanisms against network attacks. The approach is based on the simulation of cyber-attacks and cyber-protection mechanisms which combines discrete-event simulation, multi-agent approach and packet-level simulation of network protocols. A lot of different experiments were carried. They were aimed to investigate dependence of defense mechanisms effectiveness parameters from network topology and configuration, structure and configuration of attack and defense teams, attack and defense mechanisms and defense teams' cooperation. Experiments showed that team cooperation leads to the essential defense effectiveness improvement. Future work is related with more thorough investigation of effectiveness of cooperation mechanisms for different teams and inter-team interaction of agents, implementation of self-adaptation and self-learning of agents. We are planning to expand the attacks and defenses library, elaborate particular components functionalities.

Acknowledgment

This research is being supported by grant of Russian Foundation of Basic Research (Project No. 07-01-00547), program of fundamental research of the Department for Informational Technologies and Computation Systems of the Russian Academy of Sciences (contract No 3.2/03), Russian Science Support Foundation and partly funded by the EU as part of the POSITIF project (contract No. IST-2002-002314) and RE-TRUST (contract No. 021186-2).

References

[1] T. Back, D.B. Fogel, Z. Michalewicz, *Evolutionary Computation*, Vol. 1. Basic algorithms and operators, Institute of Physics Publishing, 2000.

[2] E. Charniak, R.P. Goldman, "A bayesian model of plan recognition," *Artificial Intelligence*, vol. 64, no. 1, 1993.

[3] S. Chen, Q. Song, "Perimeter-based defense against high bandwidth DDoS attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 7, 2005.

[4] P. Cohen, H.J. Levesque, "Teamwork," *Nous*, no. 35, 1991.

[5] V.V. Druzhinin, D.S. Kontorov, M.D. Kontorov, *Introduction Into Conflict Theory*, Moscow, Radio i svyas', 1989 (in Russian).

[6] T. Gamer, M. Scholler, R. Bless, "A granularity-adaptive system for in-network attack detection," *Proceedings of the IEEE / IST Workshop on Monitoring, Attack Detection and Mitigation,* 2006.

[7] 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.

[8] V. Gorodetski, I. Kotenko, "Conceptual foundations of stochastic simulation in the Internet," *Proceedings of system analysis institute of RAS*, vol.9, Moscow, URSS, 2005 (in Russian).

[9] B. Grosz, S. Kraus, "Collaborative plans for complex group actions," *Artificial Intelligence*, vol. 86, 1996.

[10] 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.

[11] P. Horn, "Autonomic Computing: IBM's Perspective on the State of Information Technology," http://www.research.ibm.com/ autonomic/manifesto/autonomic_computing.pdf

[12] J. Ioannidis, S.M. Bellovin, "Implementing pushback: router-based defense against DDoS attacks," *Symposium of Network and Distributed Systems Security (NDSS)*, California, 2002.

[13] Y. Ishida, *Immunity-Based Systems A Design Perspective,* Springer Verlag, 2004.

[14] C. Jin, H. Wang, K.G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," *Proceedings of ACM Conference on Computer and Communications Security*, 2003.

[15] J.O. Kephart, D.M. Chess, "The vision of autonomic computing," *IEEE Computer Magazine*, no. 1, 2003.

[16] A. Keromytis, V. Misra, D. Rubenstein, "SOS: secure overlay services," *ACM SIGCOMM'02*, Pittsburgh, PA, 2002.

[17] I.V. Kotenko, "Agent-based modeling and simulation of cyber-warfare between malefactors and security agents in internet," *19th European Simulation Multiconference "Simulation in wider Europe"*, 2005.

[18] I.V. Kotenko, A.V. Ulanov, "Agent-based simulation of DDOS attacks and defense mechanisms," *Journal of Computing*, Vol.4, Issue 2, 2005.

[19] I. Kotenko, A. Ulanov, "Agent teams in cyberspace: security guards in the global internet," *International Conference on CYBERWORLDS*. 2006.

[20] I. Kotenko, A. Ulanov, "Multi-agent framework for simulation of adaptive cooperative defense against internet attacks," *Lecture Notes in Artificial Intelligence*, vol.4476, 2007.

[21] V.A. Lefevre, *Reflexion*, Moscow, "Kognito-Center", 2003 (in Russian).

[22] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat, "Lessons from three views of the internet topology: technical report," *CAIDA*, 2005.

[23] J. Mirkovic, S. Dietrich, D. Dittrich, P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms*, Prentice Hall PTR, 2004.

[24] J. Mirkovic, M. Robinson, P. Reiher, G. Oikonomou, "Distributed defense against DDoS attacks," *Technical Report CIS-TR-2005-02*, University of Delaware, 2005.

[25] M. Negoita, D. Neagu, V. Palade, *Computational Intelligence Engineering of Hybrid Systems*, Springer Verlag, 2005.

[26] OMNeT++ homepage. http://www.omnetpp.org/

[27] C. Papadopoulos, R. Lindell, I. Mehringer, A. Hussain, R. Govindan, "Cossack: coordinated suppression of simultaneous attacks," *DISCEX III*, 2003.

[28] T. Peng, L. Christopher, R. Kotagiri, "Protection from distributed denial of service attack using history-based IP filtering," *IEEE Conference on Communications,* 2003.

[29] K.S. Perumalla, S. Sundaragopalan, "High-fidelity modeling of computer network worm," *20th Annual Computer Security Applications Conference (ACSAC'04)*, December 06-10, 2004.

[30] F. Silva, M. Endler, F. Kon, R.H. Campbell, M.D. Mickunas, "Modeling dynamic adaptation of distributed systems," *Technical Report UIUCDCS-R-2000-2196*, Department of Computer Science, University of Illinois at Urbana-Champaign, 2000.

[31] M. Tambe, "Towards flexible teamwork," *Journal of AI Research*, vol. 7, 1997.

[32] 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.

[33] R. Want, T. Pering, D. Tennenhouse, "Comparing autonomic and proactive computing," *IBM Systems Journal*, vol.42, no.1, 2003.

[34] M.P. Wellman, D.V. Pynadath, "Plan Recognition under Uncertainty," *Unpublished web page*, 1997.

[35] C.C. Zou, N. Duffield, D. Towsley, W. Gong, "Adaptive defense against various network attacks," *IEEE Journal on Selected Areas in Communications: High-Speed Network Security (J-SAC)*, vol. 24, no. 10, 2006.