AGENT-BASED SIMULATION OF DISTRIBUTED DEFENSE AGAINST COMPUTER NETWORK ATTACKS

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

KEYWORDS
Agent-based Simulation, Teamwork, Security, Computer network attacks, Distributed Denial of Service

ABSTRACT

The paper describes the agent-based approach and software environment (based on OMNeT++ INET Framework) developed for simulation of distributed defense mechanisms which can be deployed in the Internet for counteraction to computer network attacks. According to the approach suggested, the cybernetic counteraction of "bad guys" and security systems is represented by the interaction of different agent teams. The main components of the software environment are outlined. One of the experiments on protection against attacks "Distributed Denial of Service" is described.

1. INTRODUCTION

The problems of information security modeling and simulation are actively discussed in the long period of time. There was developed the number of different models of particular defense mechanisms and they were simulated successfully. But, as before we lack for advanced models and simulation environments that let formalize the complex antagonistic nature of information security as complicated technical-organizational process.

This paper proposes an agent-based approach and software environment for simulation of counteraction between malefactors and defense systems in the Internet represented as an antagonistic interaction of different teams of software agents. Agent-based modeling and simulation of network security in the Internet assumes that agent competition is represented as a large set of semi-autonomous interacting agents (Kotenko 2005). The *aggregate system behavior* emerges from evolving local interactions of agents in a dynamically changing environment specified by computer network model. Our approach is based on *agent teamwork frameworks* (Cohen and Levesque 1991; Fan and Yen 2004; Grosz and Kraus 1996; Kotenko and Stankevich 2002; Tambe 1997; Tambe and Pynadath 2001; Yen et al. 2001; etc.). We investigate our approach on an example of simulating defense methods against one of the most harmful classes of computer network attacks – "Distributed Denial of Service" (DDoS) (Mirkovic et al. 2004).

The idea of DDoS attack consists in reaching the global goal – the denial of service of some resource (for example Web-server) – due to joint efforts of many hosts (zombies) that are acting on attack side sending a huge number of network inquiries to the victim host (network). The main task of defense systems against DDoS is to accurately detect these attacks and quickly respond to them (Xiang and Zhou 2004). Traditional defense include detection and reaction mechanisms (Xiang et al. 2004). Adequate victim protection can only be achieved by cooperation of different distributed components (Mirkovic et al. 2005). So, the DDoS problem requires a distributed cooperative solution (Mirkovic et al. 2004; Mirkovic et al. 2005). There are a lot of architectures for distributed cooperative defense mechanisms (Chen and Song 2005; Papadopoulos 2003; Keromytis et al. 2003; Xuan al. 2002; Xiang and Zhou 2004; Mirkovic et al. 2004; etc.).

Our goal is to try to simulate different DDoS defense methods and develop the investigation environment which can help elaborate well-grounded recommendations on the choice of efficient defense mechanisms. In (Kotenko 2005) we described the ontology of DDoS attacks and defense mechanisms, presented specifications of structure of DDoS and defense agents' team, described the formal model of computer network and determined software prototypes on Visual C++ 6.0 and Java 2 and experiments with them. In this paper, based on the main ideas considered in (Kotenko 2005), we define more exactly the used agent-based approach, consider a new powerful simulation environment developed on OMNeT++ INET Framework and demonstrate the possibilities of this environment on an example of one of many experiments on protection against attacks "Distributed Denial of Service". The rest of the paper is structured as follows. *Section 2* outlines suggested agent-based 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.

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

2. AGENT-BASED APPROACH FOR SIMULATION OF DISTRIBUTED DEFENSE

The problem of multi-agent modeling and simulation of cybernetic opposition processes is represented as the task of antagonistic interaction of the agents-malefactors' team and the defense team (Kotenko 2005). The agents of different teams compete to reach the opposite intentions. The agents of one team cooperate to realize the overall intention (implementing the threat or defense of computer network).

It is offered that each team of agents is organized by the group (team) plan of the agents' actions. As result, a team has a mechanism of decision-making about who will execute particular operations. 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 during the plan execution. The communications between agents are caused by joint intentions and rules that every agent has. Pro-active and reactive communications take place in the team. 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. The team members have the shared mental model. The agents can make the "cutoff" of the team mental state due to forming the joint intentions on the different levels of abstraction. The hierarchy of intentions is jointly established by the team members to make the team reach its goal in coordinated way. This is the consequence of agents' joint responsibilities.

Let us represent *the DDoS attack system* as an agent team. The agents aim the shared goal – the realization of attack "denial of service" for some host or network. Analyzing the present DDoS methods it is possible to determine at least two types of attack components: "Daemon" executes the attack directly; "Master" coordinates the actions of other components. 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. The important parameters on this stage are agents' amount and their state of distribution. 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. 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 send the attack packets with the various rate, spoof source IP address and do it with various rates.

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. In compliance with the general approach we set the following defense agent classes: "Sensor" – agent of initial information processing; "Sampler" – the network data collector that forms the traffic model; "Detector" – attack detection agent; "Filter" – agent of attack traffic filtering; "Investigator" – agent of attack investigation.
In the initial moment of time the defense agents are deployed on hosts corresponding to their roles: sensor is deployed on the way of traffic to defended host; sampler – on any host in defended subnet; detector – on any host in defended subnet; filter – on the entrance to defended subnet; investigator – on any host beyond the subnet. The joint goal of defense team is to protect against DDoS attack. Detector watches on its accomplishing. Sensor processes information about network packets and collects statistic data on traffic for defended host.

Samplers are deployed in the defended subnet to collect the data on its normal functioning and to detect anomalies. The examples of implemented detection mechanisms are "Hop Count Filtering" (HCF) (Jin et al. 2003) and "Source IP address monitoring" (SIPM) (Peng et al. 2003). The local sub-goals of sampler implementing these methods can be as follows: sending to detector the message of its workability; network packets processing; building the table of IP addresses for HCF and the table of hops for SIPM; anomaly detection; forming and sending the messages to filter the traffic from suspicious IP addresses. Sampler builds the traffic model in the learning mode. The traffic model is based on two tables mentioned. The first consists of "approved" IP addresses, the second – of "approved" set of distances to other subnets. It is built based on the following relations: <the first 24 bits of address – the amount of hops>. When sampler is in the normal mode it analyses each incoming packet, takes the IP address and calculates the hops amount. It looks in the corresponding tables the coincidences. If one of results is negative, then sensor sends to filter the command to filter the packets coming from this IP address. Each of mechanisms has the counter of detected "malicious" addresses to compare their effectiveness.

Detector's local goal is to make the decision if the attack happens. In developed prototype the following method is realized. If detector decides that there is a DDoS attack on the basis of data from sensors and samplers. It sends its decision and N addresses of attack hosts to filter and to investigator. *Filter*'s local goal is to filter the traffic on the basis of data from detector. If it was determined that the network is under attack, then filter begins to filter the packets from the given hosts. The goal of *investigator* is to identify and defeat the attack agents. When investigator receives the message from detector, it examines the given addresses on the presence of attack agents and tries to defeat them.

3. SIMULATION ENVIRONMENT

To choose the simulation tool the comprehensive analysis of the following systems was made: NS2 (NS2), OMNeT++ INET Framework (OMNeT++), SSF Net (SSF Net), J-Sim (J-Sim) and some others. We discovered that the OMNET++ INET Framework satisfies to these requirements best of all. OMNET++ is the discrete event simulator (OMNeT++). The change of state happens in the discrete moments of time. The simulation is being held by the future event list sorted by time. The event may be the beginning of packet transmission, time-out, etc. The events occur inside the simple modules. Such modules have the functions of initialization, message processing, action (alternatively), end of work. 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.

Agents are deployed on the hosts in the simulation environment. They are installed by connecting to the modules serving transport and network layers of protocol stack simulated in OMNeT++ INET Framework. The generalized representation of agent "sampler" structure is depicted in Figure 1. Sampler contains the transport layer (depicted as a message), needed to communicate with other agents, network layer (depicted as a blue cube) to collect traffic data and agent kernel (depicted as a blue shape of human figure). The latter contains the communication language, the knowledge base and the message handlers from the neighbor modules. The representation of sampler deployment into the simulation environment is depicted in Figure 2. One can see that agent is plugged into the host through the "tcp" module. Agent is also connected with the "sniffer" module that is used to analyze the network packets.

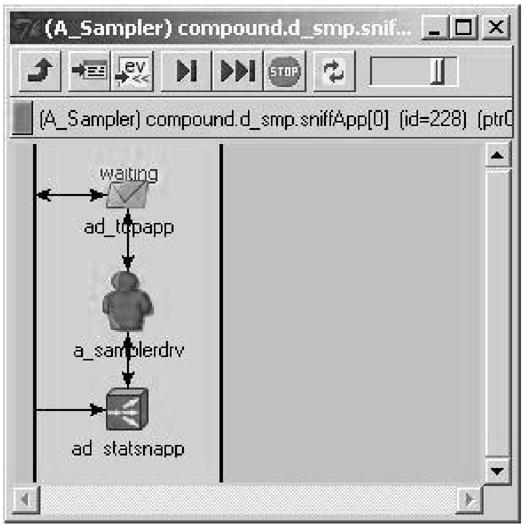

(A_Sampler) compound.d_smp.snif...
(A_Sampler) compound.d_smp.sniffApp[0] (id=228) (ptr0
waiting
ad_tcpapp
a_samplerdrv
ad_statsnapp

Figure 1. Generalized representation of agent "sampler" structure"

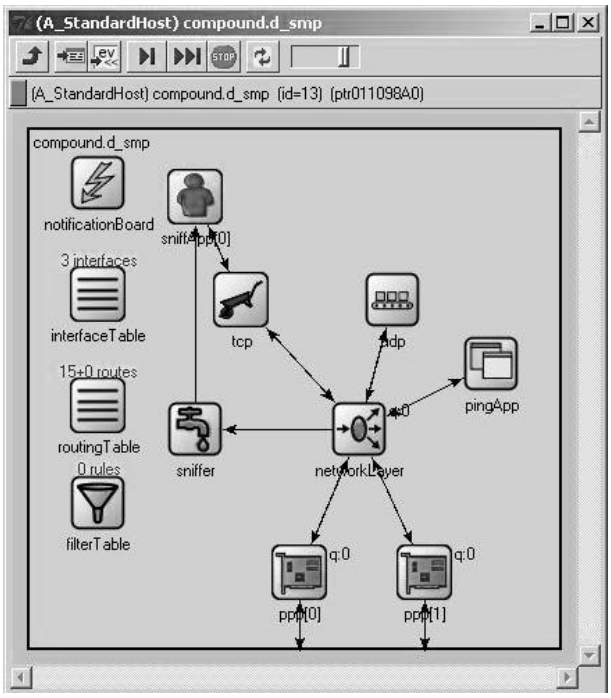
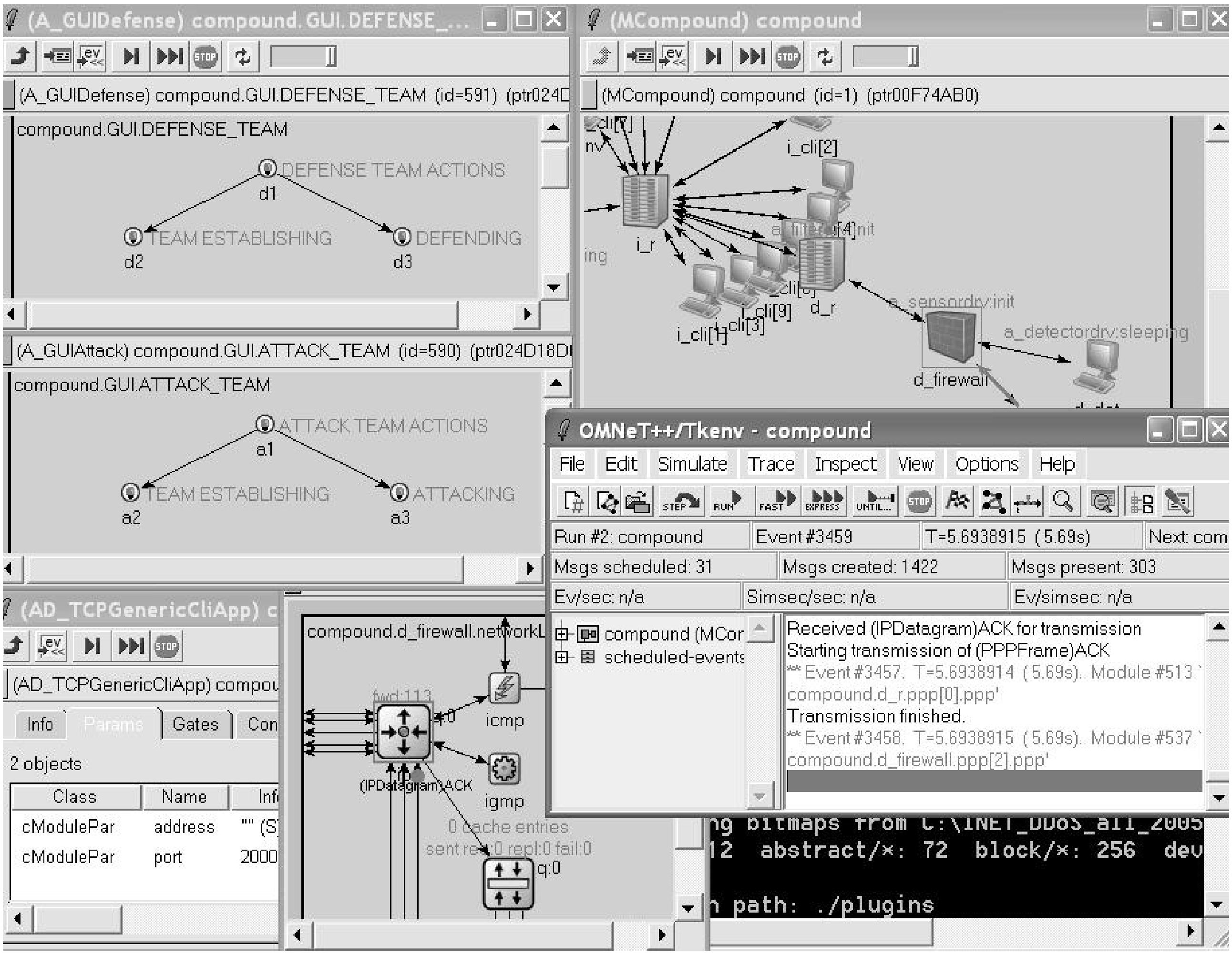


Figure 2. Representation of agent "sampler" deployment

At the basic window of visualization (Figure 3, at upper right), a simulated computer network is displayed. The network represents a set of the 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 window for simulation management (Figure 3, on the right in the



(A_GUIDefense) compound.GUI.DEFENSE_TEAM (id=591) (ptr024D

(A_GUIAttack) compound.GUI.ATTACK_TEAM (id=590) (ptr024D18D

(MCompound) compound (id=1) (ptr00F74AB0)

(AD_TCPGenericCliApp) compou

2 objects

Class	Name	Inf
cModulePar	address	"" (S
cModulePar	port	2000

```
Run #2: compound       Event #3459       T=5.6938915 ( 5.69s)     Next: com
Msgs scheduled: 31        Msgs created: 1422        Msgs present: 303
Ev/sec: n/a          Simsec/sec: n/a             Ev/simsec: n/a
```

```
Received (IPDatagram)ACK for transmission
Starting transmission of (PPPFrame)ACK
** Event #3457.  T=5.6938914 ( 5.69s).  Module #513 `
compound.d_r.ppp[0].ppp'
Transmission finished.
** Event #3458.  T=5.6938915 ( 5.69s).  Module #537 `
compound.d_firewall.ppp[2].ppp'
```

```
ng bitmaps from C:\INET_DDoS_all_2005
12  abstract/*: 72  block/*: 256  dev
n path: ./plugins
```

Figure 3. Examples of windows used during simulation process

middle) allows looking through and changing simulation parameters. It is important that it is possible to see the events which are valuable for understanding attack and defense mechanisms on time scale. Corresponding status windows (Figure 3, at upper left) show the current status of agents' teams. It is possible to open different windows which characterize functioning (the statistical data) of particular hosts, protocols and agents.

Since all simulated processes take place in the Internet, the network model should be in the heart of simulation environment. One of the examples of computer networks for simulation is represented in Figure 4. We used different configurations of computer networks. Each network is represented as a set of hosts connected by the channels. Hosts can fulfill different functionality depending on their parameters or a set of internal modules. The routers are labeled with the sign "⛁".The hosts are connected with the channels. Their parameters can be changed. They are as follows: Delay – delay of packets propagation; Datarate – the speed of packets transmission. The hosts where attack agents are deployed are red; the hosts with defense agents are green. Above the colored hosts there are the strings that indicate the corresponding state of deployed agents. The other hosts are the standard hosts that generate the generic network traffic. Each network host can consist of the following modules: ppp is responsible for the data link layer (the router can have several ppp according to the number of interfaces); networkLayer is for the network layer; pingApp is responsible for applications using ICMP; tcp serves for TCP; udp is serving for UDP; tcpApp[0] is the TCP application (there can be a number of them); notificationBoard is used for logging the events on host; interfaceTable contains the table of network interfaces; routingTable contains the routing table; filterTable contains the filtering table. The applications (including the agents) are installed on the hosts by connecting to appropriate protocol modules. 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. Hosts produce generic traffic including the traffic to defended host; (3) The subnet of attack where the attack team is deployed.

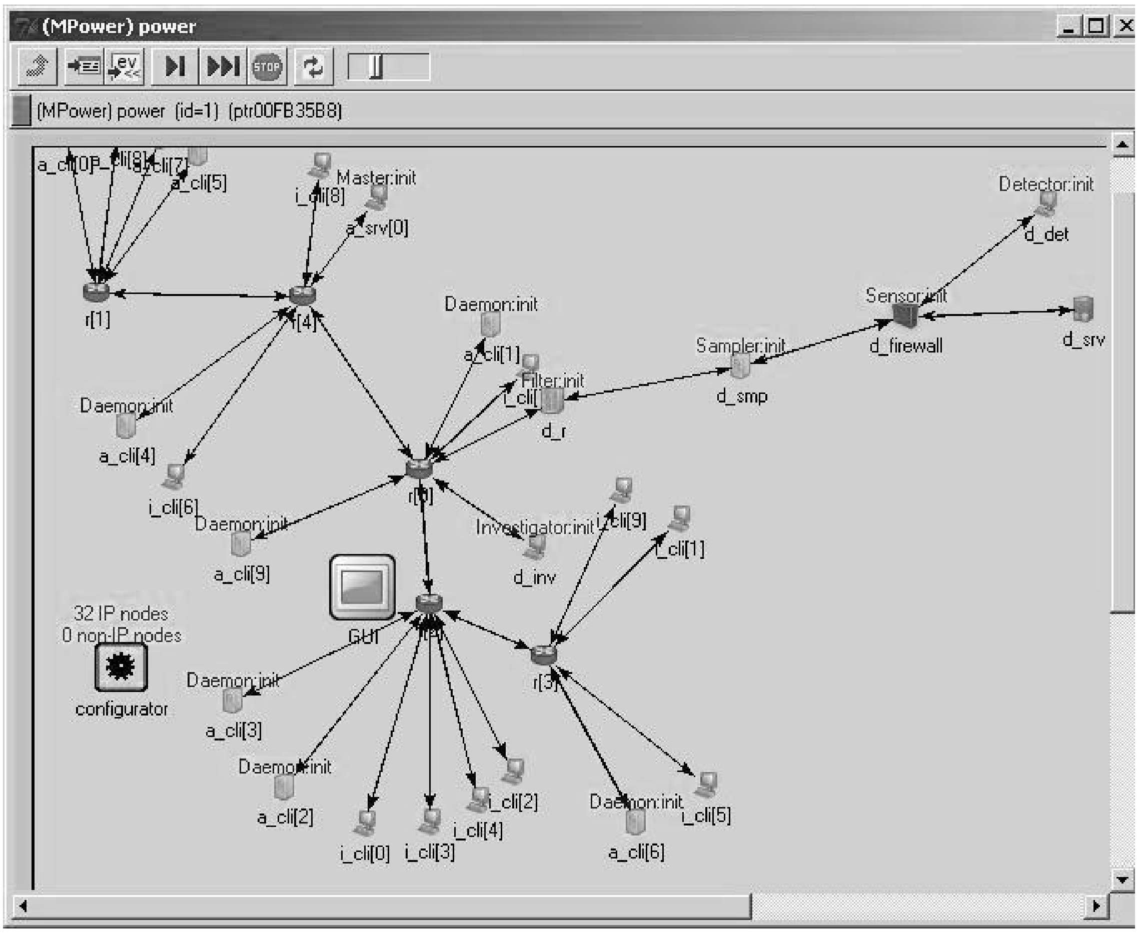


Figure 4. Example of a computer network for simulation

4. SIMULATION EXAMPLE

We are in the process of implementing simulation experiments for different cooperative active and passive defense mechanisms against DDoS attacks, including ”hop-by-hop” IP traceback, backscatter traceback, overlay networks for ip-traceback, large scale IP traceback, server roaming, congestion puzzles, change-point detection, selective pushback, aggregate based congestion control and pushback, etc.

Let us examine one of simple simulation scenarios to demonstrate possibilities of the software environment. The routers in this network are connected by fiberglass channels with bandwidth 512 Mbit. The other hosts are connected by 10 Mbit Ethernet channels.

Some time after the start of simulation, clients begin to send the requests to server and it replies. That is the way generic (normal) network traffic is generated. The formation of defense team begins some time after the start of simulation. The defense agents (investigator, sensor and filter) connect to detector. They send to detector the messages saying that they are alive and ready to work. Detector stores this information to its knowledge base. The formation of attack team occurs in the same way. The defense team actions begin after the team formation. Sensor starts to collect the traffic statistics (the amount of transmitted bytes) for every IP-address. Detector gets statistics and detects if there is an attack. Then it connects to filter and investigator and sends them the IP-addresses of suspicious hosts.

When attack actions begin, master requests every daemon if it is alive and ready to work. When all daemons were examined, it occurs that they all are workable. Master calculates the rate of attack for every daemon. Then master sends the corresponding attack command to every daemon. Daemons start the attack by sending, e.g., the UDP packets to the victim server with the given rate. Sensors and samplers send to detector the list of IP addresses and the amount of bits transmitted for the given time interval. Detector determines which hosts (IP addresses) transmit the traffic that exceeds the maximum allowable size. Detector sends these addresses to filter to apply filtering rules and to investigator to trace and defeat the attack agents. After applying the filtering rules by filter the traffic to the server was lowered. And agent-investigator tries to defeat attack agents. It succeeds to defeat two of them. The remaining daemon continues the attack. Master redistributed the attack load for it. But the attack packets do not reach the goal and are filtered at the entrance of the defended network.

5. CONCLUSION

The main results of the work we described in the paper consist in developing basic ideas on agent-based simulation of distributed defense mechanisms (on an example of protecting against attacks DDoS) and implementing corresponding software environment. According to suggested approach, the cybernetic opposition of malefactors and security systems is represented by the interaction of different teams of software agents – malefactors' team and defense team. The environment developed is written in C++ and OMNeT++ INET Framework. It allows imitating a wide spectrum of real life DDoS attacks and defense mechanisms. Different 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 building more realistic environment, and conducting experiments to both evaluate network security and analyze the efficiency and effectiveness of security policy against different attacks.

6. ACKNOWLEDGEMENT

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

Chen, S. and Q. Song. 2005. "Perimeter-Based Defense against High Bandwidth DDoS Attacks". *IEEE Transactions on Parallel and Distributed Systems*, Vol.16, No.7.

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

Fan, X. and J. Yen. 2004. "Modeling and Simulating Human Teamwork Behaviors Using Intelligent Agents". *Journal of Physics of Life Reviews*, Vol. 1, No.3.

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

Jin, C.; H. Wang; K.G. Shin. 2003. "Hop-count filtering: An effective defense against spoofed DDoS traffic". *Proceedings of the 10th ACM Conference on Computer and Communications Security*.

J-Sim. http://www.j-sim.org

Keromytis, A.D.; V. Misra; D. Rubenstein. 2003. "SOS: An architecture for mitigating DDoS attacks". *Journal on Selected Areas in Communications*, Vol. 21.

Kotenko, I. and L. Stankevich. 2002. "The Control of Teams of Autonomous Objects in the Time-Constrained Environments". *Proceedings of the IEEE International Conference "Artificial Intelligence Systems*, IEEE Computer Society.

Kotenko, I. 2005. "Agent-Based Modeling and Simulation of Cyber-Warfare between Malefactors and Security Agents in Internet". *19th European Simulation Multiconference "Simulation in wider Europe"*.

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

Mirkovic, J.; M. Robinson; P. Reiher; G. Oikonomou. 2005. "Distributed Defense Against DDOS Attacks". *University of Delaware CIS Department Technical Report CIS-TR-2005-02*.

NS2. http://www.isi.edu/nsnam/ns/

OMNeT++. http://www.omnetpp.org/

Papadopoulos, C.; R. Lindell; I. Mehringer; A Hussain; R. Govindan. 2003. "Cossack: Coordinated suppression of simultaneous attacks". *Proceedings of DISCEX III*.

Peng, T.; L. Christopher; R. Kotagiri. 2003. "Protection from Distributed Denial of Service Attack Using History-based IP Filtering". *IEEE International Conference on Communications.*

SSF Net. http://www.ssfnet.org

Tambe, M.: 1997. "Towards flexible teamwork". *Journal of AI Research*, Vol.7.

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

Xiang, Y. and W. Zhou. 2004. "An Active Distributed Defense System to Protect Web Applications from DDoS Attacks". *The Sixth International Conference on Information Integration and Web Based Application & Services.*

Xuan, D.; R. Bettati; W. Zhao. 2002. "A gateway-based defense system for distributed dos attacks in high-speed networks". *IEEE Transactions on Systems, Man, and Cybernetics*.

Xiang, Y.; W. Zhou; M. Chowdhury. 2004. "A Survey of Active and Passive Defence Mechanisms against DDoS Attacks". *Technical Report, TR C04/02*, School of Information Technology, Deakin University, Australia.

Yen, J.; J. Yin; T.R. Ioerger; M. Miller; D. Xu; R. Volz. 2001. "CAST: Collaborative agents for simulating teamworks". *Proceedings of IJCAI'2001*.

BIOGRAPHY

IGOR KOTENKO graduated with honors St.Petersburg Academy of Space Engineering and St.Petersburg Signal Academy. 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. He is leading the computer security research group in St. Petersburg Institute for Informatics and Automation. His e-mail address is ivkote@iias.spb.su and his Web-page can be found at http://space.iias.spb.su/ai/kotenko/.

ALEXANDER ULANOV graduated from St. Petersburg State Politechnical University (2004), received his master's degree (2004) in the area "System analysis and control". He is PhD student in the field of agent-based modeling and simulation for computer network attacks. His e-mail address is ulanov@iias.spb.su and his Web-page can be found at http://space.iias.spb.su/ai/ulanov/ .