Formal Framework for Modeling and Simulation of DDoS Attacks Based on Teamwork of Hackers-Agents

Igor Kotenko, Alexey Alexeev, Evgeny Man'kov
St. Petersburg Institute for Informatics and Automation, 39, 14th Liniya, Russia
{ ivkote@mail.iias.spb.su, alekha@nm.ru, toltec@pisem.net }

Abstract

The modern Internet is at rather dangerous stage of its life cycle. Taking into account a today's level of computer network security, the Internet can simply cease to work, if the current tendency of growth of number and capacity of distributed denial-of-service (DDoS) attacks to root servers will proceed. In the paper we discuss that in order to combat DDoS, the computer community needs to develop a strong theoretical basis upon which to harden information systems and infrastructures so they can survive such attacks. We introduce an agent-based formal framework for modeling and simulation of DDoS attacks. The framework and software tool developed can be used for conducting experiments to analyze computer network vulnerabilities and evaluate efficiency and effectiveness of security policy.

1. Introduction

One of the most harmful classes of attacks aiming at destruction of network resources availability is "Denial of Service" (DoS). The basic feature of "Distributed Denial of Service" (DDoS) attacks is coordinated use of enormous remote hosts-"slaves" for generation of ill-intentioned traffic [8, 9, 12]. DDoS attack is preceded by breakings of tens, hundreds or even thousands computers in which the special DDoS-software is established thus allowing to carry out coordinated DoS attacks against victim hosts. DDoS poses an immense threat to the Internet preventing legitimate users of a service from using this service.

The seriousness of the DDoS problem and the increased frequency of DDoS attacks have led to the development of numerous DDoS defensive mechanisms. Unfortunately, the existing theoretical basis that should support realization of defensive mechanisms against such class of attacks is poor. According to our opinion, among many reasons, the above is stipulated by weakness of fundamental research that consider defense against DDoS attacks as a task of adversarial competition between security systems and malefactors' attacking systems, in particular, the research intending development of adequate formal framework for exploratory modeling and respective software architecture for simulation of attacks and defensive software components of computer network.

The goals of the paper are as follows: (1) development of formal framework for modeling and simulation of a wide spectrum of DDoS attacks based on agents' teamwork; (2) elaboration of formal specifications of a representative spectrum of such attacks; (3) implementation of agent-based software tool making it possible to simulate DDoS attacks and respective responses of the attacked computer network.

The rest of the paper is structured as follows. *Section 2* outlines the common approach used for modeling and simulation of DDoS attacks. *Section 3* describes the ontology of DDoS attacks developed. *Section 4* presents the structure and common scheme of operation of DDoS agents. *Section 5* outlines the specifications of agents' plans. *Section 6* describes the formal model of the attacked computer network. *Section 7* considers the Attack simulator implementation and experiments fulfilled. *Conclusion* outlines results of the paper.

2. Framework for hackers-agents' teamwork

The agents' team realizes teamwork, if the team members 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 [2, 6, 7, 10, 11, etc.]. A set of approaches to description, formalization and simulation of the agents' teamwork is known.

For the organization of teamwork of hackers-agents realizing coordinated distributed attacks (including DDoS attacks), we have used the base ideas stated in works on joint intention theory [2], shared plans theory [6] and combined theories of agents' teamwork [7, 10, 11, etc.].

As in the joint intention theory, the basic elements allowing the team to perform a common task are common (group) intentions, but their structuring is carried out in

Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)
0-7695-1931-8/03 $ 17.00 © 2003 IEEE

the same way as the plans are structured in the shared plans theory [10, 11]. The common (group, individual) intention and commitment are associated with each node of the whole hierarchical plan; these intention and commitment are used for execution of the whole plan. Each agent needs to possess the group beliefs about other teammates. All agents' communication is managed by means of common commitments built on the basis of common intentions. For this purpose the special mechanisms of agents reasoning about communications should be used [10, 11].

The suggested technology for creation of the hackers-agents' team consists in realization of following stages:

(1) *forming the subject domain ontology*;

(2) *determining the agents' team structure* in terms of a hierarchy of group and individual roles. Leaves of the hierarchy correspond to roles of individual agents, intermediate nodes – to group roles;

(3) *defining the agents' interaction-and-coordination mechanisms* (including roles and scenarios of an agents' roles exchange);

(4) *specifying the agents' actions plans* on generation of attacks. It is offered to carry out the plan specification as a hierarchy of attribute stochastic formal grammars, connected by substitution operation. The plan hierarchy specification is carried out for each role. For group plans, joint activity should be expressed obviously. The following elements are described for each plan: (a) entry conditions; (b) conditions at which the plan stops to be executed; (c) actions which are carried out at a team level as a part of a common plan;

(5) *assigning roles and allocating plans between the agents*. Agents can exchange roles in dynamics of the plan execution. Requirements to each role are formulated as union of requirements to those parts of the plan which are put in correspondence to the role. Agents' functionalities are generated automatically according to roles;

*(*6) *realizing the teamwork by set of state-machines* built as a result of interpretation of a hierarchy of attribute stochastic formal grammars which set the plan hierarchy specification. The state machines realize a choice of the plan which will be executed and a fulfillment of the established sub-plans in a cycle "agents' actions – responses of environment".

3. Ontology of DDoS attacks

The developed ontology comprises a hierarchy of notions specifying intentions and actions of team of malefactors implementation of DDoS attacks 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 [5]. The notions of the ontology of an upper layer can be interconnected with the corresponding notions of the lower layer through one of four kinds of relationships: (1) "*Part of*" (decomposition); (2) "*Kind of*" (specialization); (3) "*Seq of*" ("Whole operation" – "Sub-operation"); (4) "*Example of*" ("type of object – sample of object").

The developed ontology includes the detailed description of DDoS domain in which high-layer notions correspond to hackers' intentions and the notions of bottom layer are specified in terms of network packets, OS calls, and audit data. In the DoS-attacks ontology developed, nodes specifying a set of software exploits for generation of DDoS attacks (Trinity V3, MSTREAM, SHAFT, TFN2K, Stacheldraht, Trin00, etc.) make up a top level of the ontology fragment. At lower levels of the fragment different classes of DoS-attacks are detailed, for example "Land" attacks (sending an IP-packet with equal fields of port and address of the sender and the receiver), "Smurf" attacks (sending broadcasting ICMP ECHO inquiries on behalf of a victim host), etc.

4. Structure and scheme of agents' operation

In the paper we limit a team of agents by three-level structure: the "client" supervises a subteam of "masters", each master manages a group of "demons", and demons (scouts and attackers) execute immediate attack actions against victim hosts.

An operation of each agent can be represented as alternation in a continuous cycle of phases (actions) on recognition of a current state, choice of action (in view of time restrictions) and its performance.

For support of teamwork of DDoS agents it is offered to use three groups of mechanisms (procedures) [10]:

(1) *maintenance of action coordination* intended for realization of coordinated initialization and termination of actions. It assumes assignment of roles to particular agents in concrete scenario, their notification about the appointed scenario and role, and reception of confirmations on their readiness to play the defined role in the scenario;

(2) *monitoring and restoration of agent functionality* 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;

(3) *maintenance of communication selectivity* which are necessary to guarantee that the benefit of the message exchange for agents' coordination surpasses a "cost" of the communication act.

5. Specification of hackers-agents' plans

The common plan of DDoS attacks implemented by team of hacker-agents has three-levels: (1) *Upper level* – intention-based scenarios of DDoS agents' team specified

in terms of time-ordered sequences of intentions and negotiation acts; (2) *Middle level* – intention-based scenarios of each agent's activity specified in terms of an ordered sequences of sub-goals; (3) *Lower level* – agent's intention realization specified in terms of sequences of low-level actions (commands).

The plan of each DDoS attack includes three main stages: preliminary, basic and final. The operations of the *preliminary stage* are 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 a set of "victim" network (host) parameters, agents-attackers begin to defeat a chosen network (host). At this time agents-scouts monitor a victim state. At detection of the attack success, agents-scouts inform other agents about this fact. In case of prevarication of a host or impossibility of defeating it, the operation is terminated or a new victim is chosen.

The formal model of attacks is specified in terms of a set of grammars interconnected through "substitution" operations [1, 3, 5]: M_A=<{G_i}, {Su}>, where {G_i} – attribute stochastic grammars, {Su} – "substitution" operations. The formal model of attack scenarios in terms of formal grammars are based on the attacks ontology described above. 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 agents' actions, and in the latter case – for simulation of attacks against real network.

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, the agents of corresponding types are installed (cloned, employed).

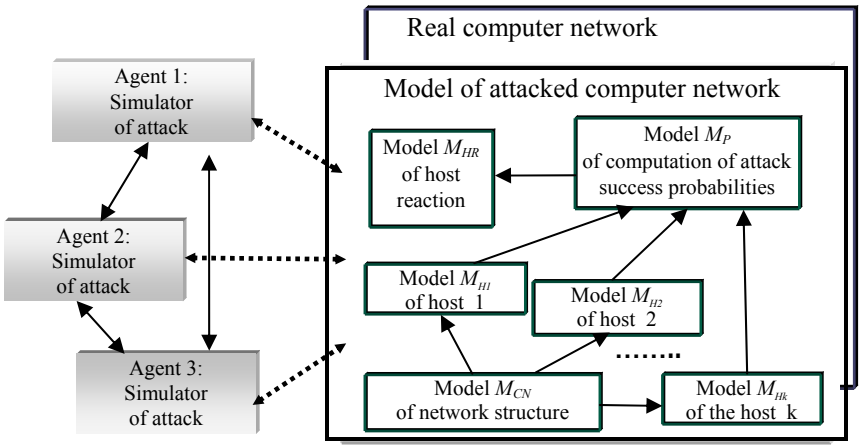

Figure 1. Conceptual view of attack simulation

Algorithmic interpretation of the attack generation specified as a family of formal generalized grammars is implemented by a family of state machines. States of each state machine are divided into three types: first (initial), intermediate, and final (marker of this state is End). The initial and intermediate states are as follows: non-terminal, those that initiate the work of the corresponding nested state machines; terminal, those that interact with the host model; auxiliary states. Transition arcs are identified with the productions of grammars, and can be carried out only under certain conditions.

6. Formal model of attacked network

An attack is fulfilled as an interactive process, in which the computer network attacked reacts on the DDoS agents' actions (Figure 1). Computer network plays the role of environment for attacker, and therefore its model must be an important part of the attack simulation tool.

Model of the attacked computer network developed can be represented as quadruple $MA = <M_{CN}, \{M_{Hi}\}, M_P, M_{HR}>$, where M_{CN} – the model of computer network structure; $\{M_{Hi}\}$ – the models of host resources; M_P – the model of computation of attack success probabilities; M_{HR} – the model of host reaction in response of attacks. The model M_{CN} of a computer network structure specifies the network address, a family of protocols used, a set of sub-networks and/or hosts of the network, connections between the sub-networks (hosts) established as a mapping matrix.

Models $\{M_{Hi}\}$ of the network host resources serve for representing the host parameters that are important for attack simulation (IP-address, type and version of OS, users' identifiers, domain names, host access passwords, domain parameters, active TCP and UDP ports and services of the hosts, running applications, etc.).

Success or failure of any attack action is determined by means of the model M_P of computation of the attack success probabilities. This model specifies the rules that determine the action success probabilities depending on the parameters of the host attacked.

The result of each attack action is determined according to the model M_{HR} of host reaction. This model is determined by a set of rules of 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.

7. Attack simulation tool implementation

The software prototype of the Attack Simulator tool is currently used for validation of the formal framework suggested and exploration of implementation issues. The engineering of the attack simulator was carried out on the basis of MASDK – Multi-Agent System Development Kit [4]. The agents generated by MASDK have the identical architecture. Differences are reflected in the content of the agents' data and knowledge bases. Each agent interacts with other agents, environment which is perceived, and, possibly, modified by agents, and user communicating with agents through the agent's interface.

The main objective of the experiments with the Attack Simulator is to evaluate the tool's efficiency for different variants of attacks and attacked network configurations. The simulation-based exploration of the Attack Simulator tool has the following purposes:

(1) Checking a computer network security policy at stages of conceptual and logical design of network security mechanisms. This type of checking is performed by simulation of attacks at a macro-level;

(2) Checking security policy of a real-life computer network. This task is performed via simulation of attacks at a micro-level, that is by generating network traffic corresponding to the real activity of malefactors.

These experiments were carried out for various parameters of the attack task specification and an attacked computer network configuration. In addition to malefactor's intention, the influence of the following input parameters on attacks efficacy was explored: degree of protection afforded by the network and personal firewall, and attacked host, and the degree of malefactor's knowledge about a network. To investigate the Attack Simulator capabilities, the following parameters of attack realization outcome have been selected: number of terminal-level attack actions, percentage of the malefactor's intentions that are successful, percentage of "effective" network responses on attack actions, percentage of attack actions blockage by firewall, and percentage of "ineffective" results of attack actions.

8. Conclusion

In the paper a formal paradigm for modeling and simulation of a broad spectrum of DDoS attacks performed by a team of DDoS agents is proposed.

The paper presents the structure of a team of agents, agent interaction-and-coordination mechanisms and specifications of hierarchies of agent plans.

The developed approach has been used for simulation-based evaluation of computer network security and analysis of both efficiency and effectiveness of security policy against DDoS attacks.

The Attack Simulator tool now supports only simulation of a wide spectrum of real-life DoS attacks. It is implemented in Visual C++ 6.0, Java 2 version 1.3.1, KQML, and XML languages.

Experiments with the Attack Simulator have been conducted, including the investigation of attack scenarios against networks with different structures and security policies.

The further development of the Attack Simulator tool will consist of enlargement of its capabilities in specification of the attack tasks, expansion of the attack classes, implementing more sophisticated attack scenarios, realizing the DDoS attacks simulation, etc.

9. Acknowledgement

This research is being supported by grant 01-01-108 of Russian Foundation of Basic Research and European Office of Aerospace R&D (Projects #1994 P).

10. References

[1] A.V.Aho, and J.D.Ullman, *The Theory of Parsing, Translation, and Compiling*, Prentice-Hall, Inc., 1972.

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

[3] K.S.Fu, *Syntactic Methods in Pattern Recognition*, Academic Press, New York, 1974.

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

[5] V.Gorodetski, and I.Kotenko, "Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool", *Lecture Notes in Computer Science*, Vol. 2516, Springer Verlag, 2002.

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

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

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

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

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

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

[12] *Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation.* SANS Institute. April 7, 2001.