Security and Embedded Systems.
D.N.Serpanos, R.Giladi (Eds.).
IOS Press. 2006.

Agent-based modeling and simulation of malefactors' attacks against computer networks

Igor KOTENKO[1], Mihail STEPASHKIN[1] and Alexander ULANOV[1]
SPIIRAS, Intelligent Systems Laboratory, Russia

Abstract. The paper analyses state of the art in modeling and simulation of attacks against computer networks and presents authors' experience in applying multi-agent technology for attack simulation. The suggested approach for attack simulation is realized via automatic imitation of distributed computer network attacks on different levels of detail. The paper describes three attack simulation tools: Agent-Based Attack Simulator, Active Vulnerability Assessment System and Agent-Based Simulator of Distributed Denial of Service (DDoS) Attacks.

Keywords. Agent-based modeling and simulation, active vulnerability assessment, computer network attacks, penetration testing

Introduction

The necessity of attack modeling and simulation has arisen since the moment of appearance of the first incidents of penetration of computers and computer networks. Using the knowledge obtained from the generalization and formalization of computer systems' vulnerabilities and cases of attacks could considerably improve the efficiency of existent protection mechanisms. This is a strong argument in favour of serious study and research of the essence and the particular features of malefactors' attacks against computer networks.

This research cannot only be restricted by the generalization of the experience; it also has to be based on formal models of attacks and attack simulation tools. These models and tools could be very valuable in the design of security systems which are capable of operating with high-level notions like "identification of an attack scenario", "attack development forecast", etc. Such capabilities could enable the system to suppress the development of an attack on-line before the irreversible consequences occur. Attack simulation tools could also play an important role in the validation of security systems and policies. Such tools could be used as testing equipment, thus cutting the costs of and decreasing the time necessary for security policy validation.

The goal of research described in the paper consists in *analysis and development of a general approach, mathematical models, ontologies and software tools intended for agent-based modeling and simulation of attacks and active analysis of computer network vulnerabilities*. We developed formal models and techniques for attack

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

modeling and simulation taking into account the malefactor's intentions, the level of their knowledge and experience, and scenarios of network attacks specified on the macro and micro levels.

The rest of the paper is structured as follows. *Section 1* reviews relevant works and outlines suggested common approach for modeling and simulation of attacks. *Section 2* describes the models and architecture implemented in Agent-Based Attack Simulator (ABAS). *Section 3* outlines the peculiarities of Active Vulnerability Assessment System (AVAS) based on main decisions realized in ABAS. *Section 4* presents the architecture of and experiments fulfilled with Agent-Based Simulator of DDoS Attacks (ABSDA). *Section 5* outlines the main results of the paper and future work directions.

1. State of the Art in Modeling and Simulation of Attacks

The research papers relevant to attack modeling and simulation can be divided into the following groups: (1) Attacks and attack taxonomies; (2) Attack languages; (3) Research immediately coupled with network attack modeling and simulation; (4) Evaluating security systems and policies, vulnerability assessment tools (scanners), signature and traffic generation tools, security metrics, etc. This list is not exhaustive. Main directions and contents of relevant works are depicted in Table 1.

Table 1. Main directions and contents of relevant works

Research directions	Main works
(1) Attacks and attack taxonomies	Lists of attack terms [5, 18]; Lists of attack categories [44]; Attack results categories [5]; Empirical lists of attack types [28]; Vulnerabilities matrices [27]; Security flaws or vulnerabilities taxonomies [25]; Taxonomies of intrusions based on the signatures [26]; Incident taxonomies [18], etc.
(2) Attack languages	CASL [2], NASL [10], CISL [13], IDMEF [8], BRO [42], Snort [46], SNP-L [55], STATL [12], GasSATA [34], LAMBDA [7], AdeLe [35], Alert correlation [40], etc.
(3) Network attack modeling and simulation	State transition analysis technique [19, 21]; Simulating intrusions in sequential and parallelized forms [4]; Cause-effect model [6]; Conceptual models of computer penetration [52]; Descriptive models of the network and the attackers [56]; Structured "tree"-based description [36, 48]; Modeling survivability of networked systems [37]; Contingency analysis based on variations in intruder attack-potential [32]; Object-oriented Discrete Event Simulation [3]; Requires/provides model for computer attacks [54]; Situation calculus and goal-directed procedure invocation [14]; Game-theoretic approaches [31]; Models of attack propagation in networks [39]; Attack graphs for vulnerability analysis [20, 50, 53]; Modeling and inference of attacker intent, objectives, and strategies [30]; Multi-stage attack analysis [9], etc.
(4) Evaluating security systems	Methodology and software tools for testing [1, 33, 43]; Evaluations of intrusion detection systems [29]; Real-time test bed [11]; Dependability models for evaluation security [38]; Penetration testing of formal models of networks for estimating security metrics [51]; Global metrics for analyzing the effects of complex network faults and attacks [17]; Knowledge-based approach to network risk assessment [49]; Model checking for analysis of network vulnerabilities [45]; Natural-deduction for automatic generation and analysis of attacks against intrusion detection systems [47], etc.

We think that an adequate approach for the investigation of remote distributed attacks on computer networks that has not been analyzed in depth is *agent-based modeling and simulation*. Our approach has applied the results of reviewed relevant works, but is evolving own theoretical and practical ideas about using formal models and multi-agent technology. We try to apply the idea that particular components of attack simulation system must be represented as a distributed system of autonomous adaptive software entities interacting via message exchange and making decisions in a cooperative and coordinated manner [22]. So, from implementation issue, a computer network attack can be considered as a sequence of coordinated actions of the spatially distributed malefactors. Each malefactor is mapped as an intelligent agent of the same architecture possessing the similar functionality. We use the teamwork interpretation of the malefactors' activity performing distributed attacks on the basis of combination of the joint intention and shared plans theories [22, 24].

The attack simulation system should be based on mechanism of automatic construction and replaying of distributed attacks scripts by combining known attacks fragments, taking into account the malefactor's intentions, his level of experience, and knowledge of computer network attacked. Functioning of the attack simulation system is specified by the *attack model* defined as hierarchical structure that consists of several levels. While describing the developed model of attacks, we defined main notions of attack generation that are formalized in the problem domain ontology "*Computer network attacks*" [16]. In the developed formal model, the basic notions of the domain correspond to malefactor's intentions and all other notions are structured according to the structure of intentions. This is a reason why the developed approach is referred to as "*intention-centric approach*". Three higher levels of the attack model correspond to an attacks scriptset, a particular script and script stages. The *attack scenarios level* defines a set of general malefactor's intentions (high level intentions or goals). This level corresponds to realization of series of scenarios which can be implemented by a group of malefactors. The *script level* defines only one malefactor's intention. The set of *script stages* can contain the following elements: reconnaissance, implantation (initial access to a host), gaining privileges, threat realization, covering tracks, and backdoors creation. Lower levels serve for malefactor subgoals refinement. The lowest level describes the malefactor's low level actions directly executing different exploits.

At the design stage, the attack simulation system operates with the *model of analyzed computer network*. This model is based on design specifications. At the maintenance stage, the attack simulation system interacts with a real *computer network*. This approach can be used at different stages of computer network life cycle, including design and exploitation stages.

2. Agent-Based Attack Simulator

The Agent-Based Attack Simulator (ABAS) [16] is built as a multi-agent system that uses two classes of agents: the agent of the first class ("*Network Agent*") simulates an attacked computer network and the second one ("*Hacker Agent*") – a hacker performing attacks against the computer network.

The agents are implemented on the basis of the technology supported by *Multi-Agent System Development Kit* (MASDK) [15]. The agents use different parts of the application ontology that is designed by use of the MASDK editor. The interaction between agents is supported by the communication component. While simulating an attack Hacker Agent sends a certain message to Network Agent. Network Agent analyzes the received message and forms a responsive message. This message is formed based on the Network Agent's knowledge base that represents the network configuration, information about possible existing attacks and reaction of the network on them. The *behaviors of the agents* specified on the basis of state-machine models, which are interpretations of behavior specified formally by use of formal grammar framework. Hacker Agent acts on the basis of nested state machines. The state machine model of Network Agent is represented by a single state machine.

The main objective of the *experiments with the prototype of ABAS* was to evaluate the tool's efficiency for simulation of different attacks. We have investigated the prototypes possibilities for realization of two tasks: (1) checking a security policy at stages of design of network security system. This task is solved by simulation of attacks at a macro-level and research of responding a network model being designed; (2) checking security policy (including vulnerabilities recognition) of a real-life computer network. This task is fulfilled by means of simulation of attacks at a micro-level, i.e. by generating a network traffic corresponding to real activity of malefactors. These experiments were carried out for various parameters of the attack task specification and an attacked computer network configuration.

3. Active Vulnerability Assessment System

The main objective of the Active Vulnerability Assessment System (AVAS) consists in finding the vulnerabilities, calculating the security metrics and determining the security level of computer network and its components [23]. *The architecture of each vulnerability assessment component of AVAS* contains the following modules: user interface; module of malefactor's model implementation; module of attack scenarios generation; module of scenario execution; data and knowledge repository; module of data and knowledge repository updating; module of security level assessment; report generation module; network interface.

The module of malefactor's model realization determines a malefactor's skill level, a mode of actions and an attack goal. *The data and knowledge repository* consists of a knowledge base (KB) about analyzed system, a KB of operation rules, and a database (DB) of attack tools (exploits). This repository contains data and knowledge which are as a rule used by malefactor when he is planning and realizing attacks. *The knowledge base about analyzed system* includes data about the architecture and particular parameters of computer network (for example, a type and a version of OS, a list of opened ports, etc) which are needed for scripts generation and attack execution. This data usually can be received by malefactor using reconnaissance actions and methods of social engineering. *The database of operation rules* contains meta- and low- level rules of "IF-THEN" type determining AVAS operation on different levels of detail. Meta-level rules define attack scenarios on higher levels. Low level rules specify attack actions based on external vulnerability database. IF-part of each rule contains (meta-) action goal and (or) condition parts. The goal is chosen in accordance with a scenario type, an attack intention and a higher level goal (specified in a meta-rule of higher level). The condition is compared with the data from database about analyzed system. THEN-part contains the name of attack action which can be applied and (or) the link on exploit. Each rule is marked with an identifier which allows us to determine the achieved malefactor's goal. *The DB of attack tools (exploits)* contains exploits and parameters of their execution. A choice of a parameter is determined by the data in KB about analyzed system. For example, the program of ftp brute force password cracking needs to know the ftp server port which can be determined by port scanning.

The module of attack scenarios generation selects the data about analyzed system from the data and knowledge repository, generates attack scriptset based on using operation (functionality) rules, monitors scriptset execution and scriptset updating at runtime, updates data about analyzed system. *The module of scenario execution* selects an attack action and exploits, prognoses a possible feedback from analyzed computer network, launches the exploit and recognizes a response of analyzed computer network.

Network interface provides: (1) in case of operation with the model of analyzed system – transferring identifiers and parameters of attacks (or network packets under more detailed modeling and simulation), and also receiving attacks results and system reactions; (2) in case of interaction with a computer network – transferring, capturing and the preliminary analysis of network traffic. The preliminary analysis includes: (1) parsing of packets according to connections and delivery of information about packets (including data on exposed flags, payload, etc.) and connections; (2) acquisition of data about attack results and system reactions, and also values of some statistics reflecting actions of AVAS at the level of network packets and connections.

The module of security level assessment is based on developed taxonomy of security metrics. It is a main module which calculates security metrics based on results of attack actions. Examples of security metrics are number of total and successful attack scenarios, number of total and successful malefactor attacks on the certain level of taxonomy hierarchy, number of attacks blocked by existing security facilities, number of discovered and used vulnerabilities, number of successful scenario implementation steps, total score of confidentiality and criticality of assets that have been successfully attacked, number of confidential and critical assets that have been successfully attacked, etc.

The module of database and knowledge repository update downloads the open vulnerability databases and translates them into KB of operation rules of low level.

The AVAS prototype was implemented and the experiments were held based on the case-study developed.

4. Agent-Based Simulator of DDoS Attacks

The Agent-Based Simulator of DDoS Attacks (ABSDA) [22, 24] is developed for *modeling and simulation of a wide spectrum of DDoS attacks and defense mechanisms* based on agents' teamwork formalism. In our approach it is offered that the agents' teamwork is organized by the team plan of the agents' actions. The mechanisms of the agents' interaction and coordination are based on three groups of procedures: (1) Coordination of the agents' actions (for coordinated initialization and termination of a common scenario); (2) Monitoring and restoring the agents' functionality; (3) Communication selectivity support (for choice of the most "useful" communications).

The prototype of ABSDA was developed by using the OmnetPP INET Framework [41] and C++. One of network fragments used for simulation is depicted in Figure 1.

On the initial phase of simulation the attack and defense teams are created. The attack team contains several agents of class "daemon" and one agent of class "master". The defense team consists of four agents ("sensor", "detector", "filter" and "investigator"). The host under defense is "d_srv". Attack team tries to fulfill the DDoS attack, and defense team protects the attacked host. ABSDA allows imitating different classes of DDoS attacks and defense mechanisms. Experiments with the prototype have been conducted, including the investigation of attack scenarios against networks with different structures and security policies.

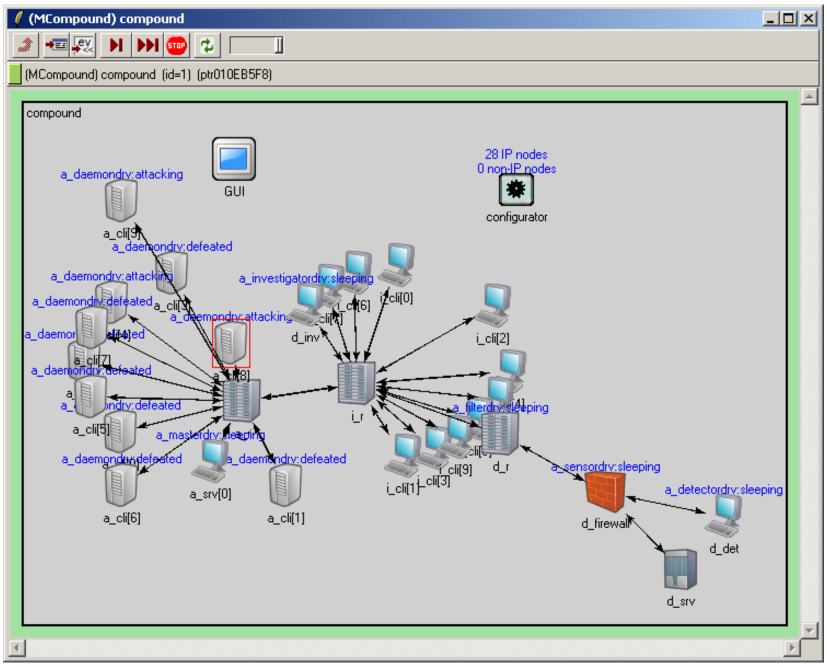

Figure 1. Example of the structure of computer network used for simulation

5. Conclusion

In the paper we described basic ideas of the agent-based modeling and simulation of network attacks. We developed the approach to be used for conducting experiments to analyze the efficiency and effectiveness of security policy against different network attacks. Software prototypes were developed. They allow imitating a wide spectrum of real life attacks. Experiments with the prototypes ware conducted, including the investigation of attack scenarios against networks with different security policies.

The further development of our modeling and simulation framework and software tools will consist of improving capabilities of the attack agents by expansion of the attack classes, implementing more sophisticated attack scenarios, and providing comprehensive experimental assessment of offered approach. Our future theoretical work is directed on development of formal basis for agent-based modeling and simulation of counteraction between attack and defense teams in the Internet.

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

[1] D.Alessandri, C.Cachin, M.Dacier, etc., Towards a taxonomy of intrusion detection systems and attacks, *MAFTIA deliverable D3, Version 1.01, Project IST-1999-11583*. (2001).

[2] Custom attack simulation language (CASL), Secure Networks. (1998).

[3] S.-D.Chi, J.S.Park, K.-C.Jung, J.-S.Lee, Network security modeling and cyber attack simulation methodology, *ACISP 2001, Lecture Notes in Computer Science, Vol.2119*. (2001).

[4] M.Chung, B. Mukherjee, R.A.Olsson, N.Puketza, Simulating concurrent intrusions for testing intrusion detection systems: parallelizing intrusions. *Proceedings of the 18th NISSC*. (1995).

[5] F.B.Cohen, Information system attacks: a preliminary classification scheme, *Computers and Security, Vol. 16, No. 1.* (1997).

[6] F.Cohen. Simulating cyber attacks, defenses, and consequences, *IEEE Symposium on Security and Privacy, Berkeley, CA.* (1999).

[7] F.Cuppens and R.Ortalo, Lambda: A language to model a database for detection of attacks, *Proceedings of RAID'2000.* (2000).

[8] D.Curry, Intrusion detection message exchange format, extensible markup language (xml) document type definition. *draft-ietf-idwg-idmef-xml-02.txt.* (2000).

[9] J.Dawkins, J. Hale, A Systematic approach to multi-stage network attack analysis, *Proceedings of the Second IEEE International Information Assurance Workshop (IWIA'04).* (2004).

[10] R.Deraison, The nessus attack scripting language reference guide, *http://www.nessus.org.* (1999).

[11] R.Durst, T.Champion, B.Witten, E.Miller, L.Spanguolo, Testing and evaluating computer intrusion detection systems, *Communications of ACM, 42(7).* (1999).

[12] S.T.Eckmann, G.Vigna, R.A.Kemmerer, STATL: An attack language for state-based intrusion detection, *Proceedings of the ACM Workshop on Intrusion Detection.* (2000).

[13] R.Feiertag, C.Kahn, P.Porras, D.Schnackenberg, S.Staniford-Chen, B.Tung, A common intrusion specification language (cisl), *Specification draft, http://www.gidos.org.* (1999).

[14] R.P.Goldman, A Stochastic model for intrusions, *Recent Advances in Intrusion Detection. Fifth International Symposium, RAID 2002, Lecture Notes in Computer Science, V.2516.* (2002).

[15] V.Gorodetski, O.Karsayev, I.Kotenko, A.Khabalov, Software development kit for multi-agent systems design and implementation, *Lecture Notes in Artificial Intelligence, Vol.2296.* (2002).

[16] V.Gorodetski, I.Kotenko, Attacks against computer network: formal grammar-based framework and simulation tool, *Recent Advances in Intrusion Detection. Fifth International Symposium. RAID 2002. Lecture Notes in Computer Science, Vol.2516.* (2002).

[17] S.Hariri, G.Qu, T.Dharmagadda, M.Ramkishore, C. S.Raghavendra, Impact analysis of faults and attacks in large-scale networks, *IEEE Security & Privacy, September/October.* (2003).

[18] J.D.Howard, T.A.Longstaff, A common language for computer security incidents, *SANDIA Report, SAND98-8667.* (1998).

[19] K.Iglun, R.A.Kemmerer, P.A.Porras, State transition analysis: a rule-based intrusion detection system, *IEEE Transactions on Software Engineering, 21(3).* (1995).

[20] S.Jha, O.Sheyner, J.Wing, Minimization and reliability analysis of attack graphs, *Technical Report CMU-CS-02-109, School of Computer Science, Carnegie Mellon University.* (2002).

[21] R.A.Kemmerer, G.Vigna, NetSTAT: a network-based intrusion detection approach, *Proceedings of the 14th Annual Computer Security Applications Conference*, Scottsdale, Arizona. (1998).

[22] I.Kotenko, Agent-based modeling and simulation of cyber-warfare between malefactors and security agents in Internet, *19th European Simulation Multiconference. ESM'05.* (2005).

[23] I.Kotenko, M.Stepashkin, Analyzing vulnerabilities and measuring security level at design and exploitation stages of computer network life cycle, *MMM-ACNS-05, Lecture Notes in Computer Science, Springer Verlag, Vol.3685.* (2005).

[24] 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 (SASEMAS '05). Utrecht, The Netherlands.* (2005).

[25] I.V.Krsul, Software vulnerability analysis, *Ph.D. Dissertation, Computer Sciences Department, Purdue University, Lafayette, IN.* (1998).

[26] S.Kumar, E.H.Spafford, A software architecture to support misuse intrusion detection. *Technical Report CSD-TR-95-009. Purdue University.* (1995).

[27] C.E.Landwehr, A.R.Bull, J.P.McDermott, W.S.Choi, A taxonomy of computer security flaws, *ACM Computing Surveys, Vol. 26, No. 3.* (1994).

[28] U.Lindqvist, E.Jonsson, How to systematically classify computer security intrusions. *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Los Alamitos, CA. (1997).

[29] R.Lippmann, J.W.Haines, D.J.Fried, J.Korba, K.Das. The 1999 DARPA off-line intrusion detection evaluation, *RAID'2000*, *Lecture Notes in Computer Science, Vol.1907.* (2000).

[30] P.Liu, W.Zang, Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Transactions on Information and System Security, Vol. 8, No. 1.* (2005).

[31] K.Lye, J.Wing, Game strategies in network security, *International Journal of Information Security, February.* (2005).

[32] J.McDermott, Attack-potential-based survivability modeling for high-consequence systems, *Third IEEE International Workshop on Information Assurance, College Park, MD, USA.* (2005).

[33] J.McHugh, The 1998 Lincoln Laboratory IDS evaluation: a critique, *RAID'2000, Lecture Notes in Computer Science, Vol. 1907.* (2000).

[34] L.Me.Gassata, A genetic algorithm as an alternative tool for security audit trails analysis, *Proceedings of the first international workshop on the Recent Advances in Intrusion Detection (RAID'98).* (1998).

[35] C.Michel, L.Me, ADeLe: an attack description language for knowledge-based intrusion detection, *Proceedings of the 16th International Conference on Information Security*. Kluwer. (2001).

[36] A.P.Moore, R.J.Ellison, R.C.Linger, Attack modeling for information security and survivability, *Technical Note CMU/SEI-2001-TN-001. Survivable Systems.* (2001).

[37] S.D.Moitra, S.L.Konda, A simulation model for managing survivability of networked information systems, *Technical Report CMU/SEI-2000-TR-020.* (2000).

[38] D.M.Nicol, W.H.Sanders, K.S.Trivedi, Model-based evaluation: from dependability to security, *IEEE Transactions on Dependable and Secure Computing. Vol.1, N.1.* (2004).

[39] S.Nikoletseas, G.Prasinos, P.Spirakis, C.Zaroliagis, Attack propagation in networks, *Theory of Computing Systems,* 36. (2003).

[40] P.Ning, D.Xu, C.G.Healey, R.A.St.Amant, Building attack scenarios through integration of complementary alert correlation methods, *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS '04).* (2004).

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

[42] V.Paxson, Bro: A system for detecting network intruders in real-time. *Proceedings of the 7th Usenix Security Symposium.* (1998).

[43] N.Puketza, M.Chung, R.A.Olsson, B.Mukherjee, A software platform for testing intrusion detection systems, *IEEE Software,* Vol.*14*, No 5. (1997).

[44] M.Ranum, A Taxonomy of Internet Attacks, *Web Security Sourcebook. John Wiley & Sons.* (1997).

[45] R.W.Ritchey, P.Ammann, Using model checking to analyze network vulnerabilities, *Proceedings SOOO IEEE Computer Society Symposium on Security and Privacy.* (2000).

[46] M.Roesch, Snort - lightweight intrusion detection for networks, *Proceedings of the USENIX LISA'99 conference.* (1999).

[47] S.Rubin, S.Jha, B.P.Miller, Automatic generation and analysis of NIDS attacks, *20th Annual Computer Security Applications Conference (ACSAC)*, Tuscon, Arizona. (2004).

[48] B.Schneier, Attack trees: modeling security threats, *Dr. Dobb's Journal, December.* (1999).

[49] B.Shepard, C.Matuszek, C.B.Fraser, etc., A Knowledge-based approach to network security: applying Cyc in the domain of network risk assessment, *The Seventeenth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence (IAAI-05), Pittsburgh, Pennsylvania.* (2005).

[50] O.Sheyner, J.Haines, S.Jha, R.Lippmann, J.M.Wing, Automated generation and analysis of attack graphs, *Proceedings of the IEEE Symposium on Security and Privacy*. (2002).

[51] S.Singh, J.Lyons, D.M.Nicol, Fast model-based penetration testing, *Proceedings of the 2004 Winter Simulation Conference.* (2004).

[52] A.J.Stewart, Distributed metastasis: a computer network penetration methodology, *Phrack Magazine, Vol 9, Issue 55.* (1999).

[53] L.Swiler, C.Phillips, D.Ellis, S.Chakerian, Computer-attack graph generation tool, *Proceedings DISCEX '01: DARPA Information Survivability Conference & Exposition II.* (2001).

[54] S.J.Templeton, K.Levitt, A requires/provides model for computer attacks, *Proceedings of the New Security Paradigms Workshop.* (2000).

[55] E.Turner, R.Zachary, Securenet pro software's snp-l scripting system, *White paper. http://www.intrusion.com, July.* (2000).

[56] J.Yuill, F.Wu, J.Settle, F.Gong, R.Forno, M.Huang, J.Asbery, Intrusion-detection for incident-response, using a military battlefield-intelligence process, *Computer Networks*, No.34. (2000).