ISSN 0147-6882, Scientific and Technical Information Processing, 2012, Vol. 39, No. 2, pp. 120–126. © Allerton Press, Inc., 2012.
Original Russian Text © D.A. Khorkov, 2012, published in Nauchno-Technicheskaya Informatsiya, Seriya 1, 2012, No. 6, pp. 9–15.

# Methods for Testing Network-Intrusion Detection Systems

**D. A. Khorkov**

*Intellectual Systems and Information Safety Regional Scientific-Educational Center,*
*Institute of Mathematics and Computer Sciences, Ural Federal University*
*e-mail: dimkhor@uralweb.ru*

Received February 24, 2012

**Abstract**—The key methods for testing network-intrusion detection systems are considered. The advantages and drawbacks of different testing methods are analyzed. The main prerequisites for developing a statistically consistent technique for testing and certification of network intrusion detection systems are formulated. The conclusion is made that it is necessary to develop a model of a complex computer attack that could be used for synthesis of the network traffic of the attacking action.

*Keywords*: computer attacks, network intrusion detection systems, computer-attack model, testing technique, network traffic

**DOI:** 10.3103/S0147688212020128

## INTRODUCTION

Protection of automated systems from computer attacks is one of the key directions in the field of informational safety. The main instruments for fighting computer attacks are intrusion detection systems (IDSs), viz., software or hardware–software complexes that allow one to reveal the actions (states) that threaten the safety of automated systems. The detection systems that use the analysis of the available network traffic as the key method for detecting attacks are called network-based. Modern IDSs are constantly modified due to the development of new algorithms and intrusion-detection methods. As a result, the problems of the analysis and substantiated selection of various IDSs (as well as the interrelated problem of IDS testing and certification) are becoming more and more complicated.

One of the first works [1] that was devoted to testing intrusion-detection systems considered a technique that is based on the known methods for software testing.

The technique consists of several tests that are aimed at evaluating IDS efficiency for the detection of known computer attacks, the effectiveness of the use of the resources of the computer the detection system is installed on, and the stability of system operation under critical conditions (in particular, with increased network loads). The field of the technique application is limited by testing network intrusion-detection systems based on the signature analysis technique. The actions of legitimate users and the attack impacts are reproduced by simulating the user's command line input using software that was specially developed in UNIX. IDS efficiency for revealing intrusions was tested using several types of attack, including an attempt to guess a password and transfer of a password file via a network. The technique was applied for testing the Network Security Monitor (NSM) intrusion-detection system, which was developed at the University of California at Davis (USA); the results of the test are presented in this paper. The authors [1] did not make any attempt to test other IDSs, such as NIDES, NADIR, or the DIDS.

## DARPA PROJECT

In 1998, in the framework of large-scale studies that were sponsored by the Defense Advances Research Project Agency (DARPA), the specialists of the MIT Lincoln Laboratory (MIT/LL) developed a project that includes a technique for testing intrusion-detection systems and a corresponding software–hardware complex. The project was called the 1998 DARPA Off-Line Intrusion Detection Evaluation; its main results were published in [2]. The project authors proposed to test intrusion-detection systems using a massif of network traffic that simulates the operation of a large computational network that has access to the Internet and contains the consequences of different computer attacks.

The required network traffic was obtained by the authors of [2] using a specially developed experimental test bed (Fig. 1), which included three groups of computers, namely, victim PCs, attacking PCs, and those intended for generation of "background" network traffic. Three computers, which ran SunOS, Linux, and Solaris, served as the targets for these attacks. An isolated PC (Fig. 1, upper left) generated network traffic to simulate the operation of several hundreds of workstations and personal computers in the intranet.

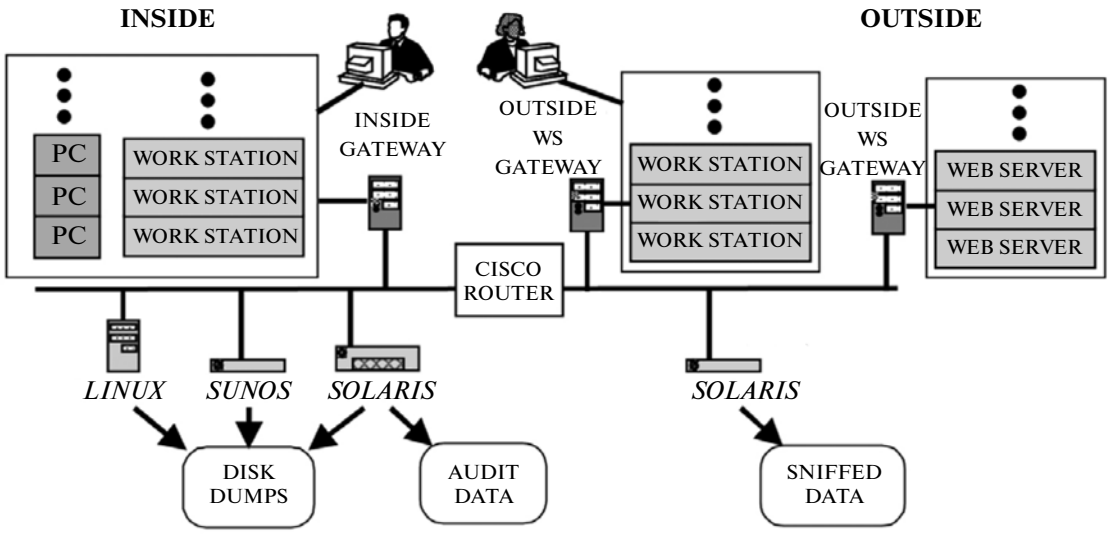


**Fig. 1.** The test bed for IDS testing (MIT/LL, 1998) [2].

Two more computers (Fig. 1, upper right) were used for generating the traffic of the external network to simulate the operation of several thousands of workstations and webservers.

It was mentioned in [2] that all attacks were launched from the external network; all in all, 32 different attacks were prepared and multiply implemented. The attacks selected by the project authors were described in detail in [3]; the classification of the computer attacks the authors made use of was taken from [4].

The thus-obtained network traffic, which contained a mixture of different packets of network attacks and "background" traffic, was recorded for 10 weeks using an analyzer (denoted as the "sniffer" in Fig. 1); it was then sent to the input of the tested IDSs. Prior to this, in isolated TCP sessions, UDP and ICMP packets were identified in the recorded traffic and numbered, with those containing attacks being specially labeled. The information on these sessions and packets was used for determining the correctness of alarm signals, as well as for calculating the probabilities of correct detection and false alarms.

The described testing method, which was based on the reproduction of the computer-attack traffic together with the "background" network traffic, can be called classical in the sense that it allows one to provide multiple recurrences of the experimental conditions. Nevertheless, the details of the implementation of this technique at MIT/LL led to criticism, as demonstrated in [5].

First, the authors of the project [2] did not specify the criteria for the realism of the synthesized "background" traffic. For example, it is stated that network traffic that does not contain attacks can yield false alarms from the intrusion detection system, but their nature was not studied and was not taken into account when synthesizing the "background" traffic. Thus, a conclusion on the probability of false alarms as applied to real network traffic cannot be made based on the IDS testing that was performed using synthesized "background" traffic.

Second, the authors of [2] do not evaluate the dependability of the IDS-operation efficiency on the intensity of the network traffic that was used in testing. Moreover, such dependability does exist even in modern software–hardware IDSs that are intended for operation with dataflows with intensities of several Gb per second.

One more weak point that was discussed in [2] is the classification of the computer attacks used by the authors. All attacks were grouped into four classes: denial-of-service, remote-to-local, user-to-root, and surveillance/probing [4]. Without taking the completeness and consistency of this classification into account, one can state that this classification is not related to the methods that are used by an IDS for detecting attacks; thus, separate evaluation of the probabilities of detecting attacks of different classes is not warranted and yields overloading of the test results with unnecessary data.

The technique for evaluating the efficiency of detection-system operation is beneath criticism and needs to be considered separately. The receiver operating characteristic (ROC), which reflects the dependence of the probability of the correct detection of the attacks of a certain class on the number of false alarms

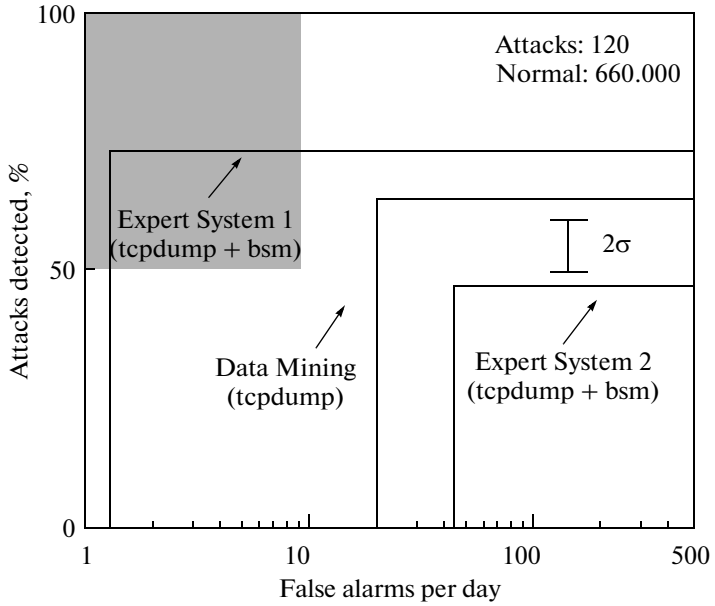


**Fig. 2.** The operating characteristics of an IDS [2].

was plotted for each IDS based on the results of its testing.

One such characteristic is shown in Fig. 2.

An operating characteristic is a vivid demonstration of the ability of a detector (receiver) to distinguish a signal against the background noise. A classical technique for plotting the operating characteristics of a receiver presumes fixation of the signal-to-noise ratio and continuous (or discrete) variation of some threshold value which, in its turn, determines the ratio of the probabilities of a false alarm and correct detection (see, e.g., [6]). When the threshold value varies continuously, the obtained operating characteristics will be continuous, in the case of a discrete threshold variation the operating characteristics will be discrete, i.e., the domain of the determination of the operating characteristics will be limited by the values that the threshold can take on. If the threshold value cannot be adjusted, the operation characteristic at the given signal-to-noise ratio will consist of a single point that is determined by the algorithm of the detector function. This case is typical of signature systems of intrusion detection, whose sensitivity cannot be adjusted. Addition of the extreme points (0,0) and (1,1) of the curve, which correspond to cases when the solution of detection is either always rejected or always accepted, does not provide any additional information on the efficiency of the detector. Clearly, the usefulness of an operating characteristic that consists of a single point is very low; it is efficient to use it when the algorithm of detector operation presumes the adjustment of a threshold value.

When plotting the operating characteristic of signature IDSs, the authors of the DARPA project made two mistakes that, to our thinking, are significant. First, the number of false alarms of the system for a certain period of time is plotted along the X-axis of the operating characteristic curve, rather than the false alarm probability. In this case, an essential parameter of the test, viz., the intensity of the background network traffic, is not mentioned. As a result, the operating characteristic loses its information value: it cannot be used for calculating the number of false alarms per unit of time if the loads in the network are different from those that were specified in the test conditions. If we turn to detection theory, the intensity of the background traffic can be considered as an analog of the noise intensity and the ratio of the intensities of the attacks and the background traffic can be taken as a signal-to-noise ratio. In this case, it becomes clear that the operating characteristic of an IDS should be plotted at the fixed (and explicitly mentioned) intensities of the attacks and the background traffic.

The second mistake is related to the fact that the authors of the project in [2] did not take the discreteness of the operating characteristic into account and represented it as a step curve (see Fig. 2), suggesting using the area of the figure under the characteristic as the key criterion of IDS efficiency. This seems to have been done for comparison of the efficiency of the signature IDSs and IDSs operation based on the principle of anomaly detection, i.e., IDSs with a discrete and continuous operating characteristic, respectively. Later, this was criticized in [7], whose authors suggested another approach for the comparison of IDS efficiency that was based on the notion of the cost of consideration, which is more correct from the standpoint of detection theory. Representation of the dis-

crete operating characteristic by a continuous curve makes the erroneous impression that, e.g., the "Expert System" detection system in Fig. 2 can detect about 75% of all attacks, which generate from 2 to 500 false alarms per day. In fact, the operating point is 75% of the detected attacks and not more than two false alarms per day

In 1999, the MIT/LL project was supplemented with a number of new attacks, including those against the operation systems of the Windows NT family. Corresponding changes were introduced into the test bed that was intended for network traffic generation. The project was called the 1999 DAPRA Off-Line Intrusion Detection Evaluation; its main results were published in [8]. Special attention was paid to the preparation of "stealth attacks," which represent the versions of known attacks that were implemented so as to hinder their detection. An attempt to analyze the potentialities for detecting "novel" attacks, i.e., those not in the IDS database, was undertaken.

As in [2], the testing was accomplished by reproducing the network traffic that was generated and recorded using the developed test bed. The massif of the network traffic consisted of two parts, namely, a 2-week fragment that did not contain any attacks and a fragment that was recorded during 1 week that contained some attacks. The exact positions of all attacks in the massif were mentioned; moreover, the traffic was labeled using a procedure that was simplified as compared to that of the 1998 project (individual TCP-connections and UPD and ICMP packets were not singled out). The time during which the attacking and the victim units exchanged network packets (plus 60 sec before and after the exchange) was considered to be the time of the attack. The attack was thought to be successfully detected if the IDS send a warning during the attack and correctly detected the IP address of the attacker. All warnings generated for other instances of the attack were considered to be false alarms.

All in all, 18 intrusion detection systems were tested. The authors demanded from the developers that each warning that was generated by the detection system contain a number in the range from 0 to 1 to represent the estimate of the probability of the attack at this instant. When several warnings were generated during one attack, the estimate with the maximum value was used.

To make it possible to adjust the number of false alarms, the specialists at MIT/LL suggested the rejection of all attacks whose probability estimate was smaller than some threshold value. The analysis of the results was two-staged: at the first stage, the capability of the IDS to detect the attacks irrespective of the number of false alarms was tested, while at the second stage the test sensitivity was set so that not more than ten false alarms were generated per day; then the fractions of the detected attacks, including those belonging to different classes, were compared. The authors of the project did not present the testing results in the form of an operating characteristic: all results were presented as tables or diagrams.

In addition to the MIT/LL 1998/1999 DARPA Off-Line Intrusion Detection Evaluation project, DARPA sponsored an analogous project for the Air Force Research Laboratory (AFRL). The main results of this work were given in [9]. Comparison of the two projects shows that they were almost identical. The distinctions concern another, more complex structure of the simulated network, the list of the attacks that were used, their classification based on the character of the network connections, and the IDS operating characteristics, which were proximate to the canonic form.

The classification of attacks that was proposed in [9] presumes their division in accordance with two classification criteria, namely, attack topology and attack complexity.

From the standpoint of the topology criterion, four classes of attacks were distinguished: local host attacks, local subnet attacks, local domain remote subnet attacks, and foreign domain attacks.

From the standpoint of their complexity, the attacks are subdivided as follows: one-to-one, one-to-many, many-to-one, and many-to-many.

Thus, the classification is focused on revealing the essential characteristics of the network connections and data flows. However, its application to the problem of IDS testing is not explained; moreover, judging by the results given in [9], this classification is not used in this work at all.

As in [2], when plotting the operation characteristic the authors of [9] did not mention the intensity of the background traffic and the total number of the launched attacks. In addition, the operating characteristics that were initially discrete were subject to piecewise-linear approximation (Fig. 3), which contradicts the principles of operation of intrusion-detection systems and misleads people who are not specialists.

The large number of common details in the work of AFRL and MIT/LL and the fact that all of them were sponsored by DARPA allows one to refer to them as one large-scale project (hereinafter, DARPA).

The analysis of the later publications that were devoted to the testing of intrusion-detection systems shows that the DAPRA project became the base for a great number of further investigations and developments in this field. It revealed the key difficulties that are faced by the developers of their own IDS-testing techniques and the typical mistakes that were made in the process of testing.

## MODERN METHODS FOR IDS TESTING

On the whole, the testing method that was suggested in the framework of the DAPRA project has a

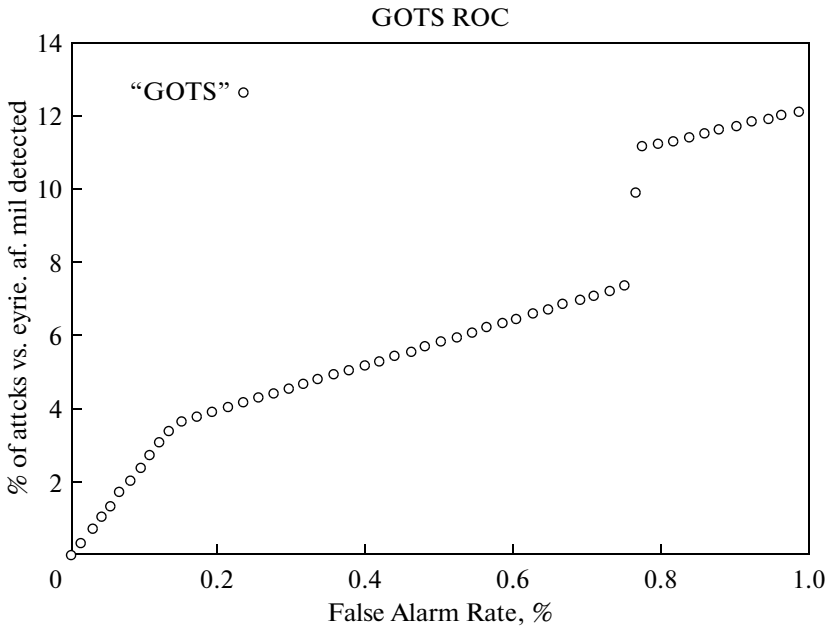


**Fig. 3.** The operating characteristic of the receiver (AFRL, 1999) [9].

number of advantages. Since from the standpoint of a network IDS a computer attack is a manifold of time-ordered network level packets (IP-packets) with a certain content, testing, in the long run, is to be reduced to the generation of the network traffic of the attacking action together with the background network traffic. The reproduction of a pre-recorded network traffic of computer attacks at the special test bed the tested IDS is connected to allows multiple repetition of the experimental conditions such that the traffic of the computer attacks will be reproduced identically each time. This is of particular importance when it is necessary to perform the comparative testing of intrusion-detection systems. The structure of the test bed for IDS testing can be maximally simplified (in the extreme case, to one computer with several network adapters and specialized software for reproduction of the attack traffic and the background traffic).

On the other hand, the requirement for the objectivity of the test results presumes that the network-traffic massifs must possess statistical variability within the specified conditions of the functioning of the victim AS. This variability can be provided by the methods of network-traffic synthesis or the method of introducing corrections to a prepared traffic massif.

The authors of [10] used the testing method suggested by DARPA and developed a software solution for the synthesis of network traffic that contains computer attacks. Based on the VMware software, they developed a hardware–software complex that allows the automated implementation of computer attacks, recording of the network traffic that is transmitted during their launching, and labeling of this traffic to automate its subsequent testing.

The use of virtual VMware machines made it possible to automatically recover the victim systems using a backup copy after the attack in the shortest possible time. The virtual network of the complex consists of a virtual machine of the attacker, a virtual victim machine, and several auxiliary virtual machines that serving as the DNS and mail servers. In addition, it includes an archive of the backup copies of the virtual machines that are used as the victim and auxiliary machines, an archive of the exploited programs that are loaded by the attacking virtual machine, and a coordinator program that governs the configuration of the virtual network and is responsible for launching the computer attacks and recording the network traffic. It should be noted that before each attack the configuration of the virtual network on which this attack can be implemented is restored to "zero."

The drawback of the suggested approach is the limitation of the intensity of the attacks, in particular, the impossibility of launching several attacks simultaneously. The advantage of this system is associated with the ability to launch "stealth" attacks using alternative methods of text encoding and fragmentation of packets (a Fragrouter and Whisker software were used).

The authors of [10] suggested the evaluation of characteristics of intrusion-detection systems, such as the number of detected and missed attacks and the number of false alarms of the system.

This technique was applied for testing two open IDSs, namely, Snort 2.3.4 and Bro 0.9a9. During the testing, 124 different attacks and 108 configurations of the victim systems were used. It should be noted that

the problem of the "background" traffic synthesis was not solved by the authors, and the attacks were not classified. Nothing was said about how the attack to be launched in the process of testing was selected or how the time lag between the attacks was determined.

Another well-known and widely used technique for testing intrusion-detection systems is reproduction of the typical conditions of IDS operation within an isolated network followed by the launching of real computer attacks in it. This testing method was used, e.g., in [10], [11]. Its apparent advantage is related to the fact that the attacking action is extremely realistic since the testing is performed using operating software modules that launch the attacks. As a rule, an operator can be personally convinced of how successful some attack was.

The software modules that launch computer attacks usually allow variation of attack parameters such as the IP-address of the attacker and victim machines, and the numbers of TCP- and UDP-ports used. In some cases, it is possible to change the setting of the packet fragmentation, text coding used in the requests, and the shell code that is send to the victim computer [11].

The main drawback of this testing method is an engineering problem that is associated with the complexity of developing a test network that consists of a large number of computers with vulnerable operation systems and applied software installed on them. In addition, it is necessary to restore the initial state of the victim computer each time after the attack in launched: otherwise, it will be possible to launch only one attack of each type. The problem of multiple recurrence of the experimental conditions (with this being one of the key requirements for test objectivity) needs special consideration.

The techniques for the testing and certification of intrusion-detection systems that were developed later [12, 13] are based on the joint use of the above-described testing methods; they are focused, first of all, on evaluation of the efficiency and functionality of IDSs. The model of a network attack that is accepted in these techniques is reduced to a set of single exploitations of well-known vulnerabilities (as a rule, in the operation systems of the Microsoft Windows family) launched by an operator or a special software. As a result, the techniques do not contain any procedures that allow one to evaluate the capabilities of the tested systems for the detection and identification of thoroughly planned multistage (complex) computer attacks, as well as "new" attacks whose signatures are absent in the IDS database.

## THE SIMULATION OF COMPUTER ATTACKS DURING IDS TESTING

We think that the most promising method for testing the ability of an IDS ability to detect complex computer attacks is the controlled synthesis of the network traffic of the attacking action in accordance with the developed model of a complex computer attack.

The model should take the fact into account that a complex computer attack consists of a sequence of elementary attacking actions (EAAs), namely, individual sets of actions by an attacker that are implemented using various software systems that are aimed at reaching the final goal of the complex attack. A criterion that unites the actions of an attacker using EAAs is a certain type of vulnerability, e.g., the vulnerability of the buffer overflow in the network service, or the application of a specific software tool, e.g., software for guessing passwords. We shall assume that provided that a significant result is preserved within the framework of the attack, the simulation of the attacking action is indivisible.

At each stage of an attack, the attacker makes a decision concerning the selection of some EAA, taking the final goal of the attack into account, the expected EAA result, and the available information on the victim AS. Each elementary attacking action necessitates the fulfillment of a number of requirements, e.g., an open TPC port, or the vulnerability of the buffer overflow in the corresponding software. Some EAAs require that the attacker have knowledge about user entries (names, passwords, etc.). An attacking action has consequences, which can manifest themselves in changes in the parameters of the victim system and in the attacker gaining knowledge about these parameters.

A computer attack develops in time and some of its stages can be characterized by both synchronism (succession) and asynchronism (parallelism). Most of the time delays during the attack have a stochastic nature depending, in particular, on such factors as the time of transfer of network packets between the attacker and the victim machines.

Thus, a key requirement for the model of a complex computer attack is the possibility of its description using an algorithm of the attacker actions, some alternatives of attack development, and the random nature of the synthesized delays. In addition, the model should allow the description of a victim automated system as a set of parameters that vary when EAAs are implemented.

Analysis of information sources showed that the existing models of computer attacks are predominantly focused on the development of algorithms for the detection of some types of attacks rather than on testing intrusion-detection systems; thus, they cannot be applied for the synthesis of the network traffic of an attacking action. The theoretical-graph approach that is widely used for describing computer attacks makes it possible to repel a possible sequence of attacks, multistage ones included, but it does not contain mechanisms for controlled branching and the simulation of dynamic systems. Deterministic models based on

FSMs contain a dynamic component and they can describe the controlled transition of a system from one state to another. However, they do not allow one to synthesize random delays and perform random selection of attack development alternatives. In contrast, stochastic models, such as probabilistic automatons, are inconvenient for use in systems that operate in accordance with a certain algorithm.

A reasonable compromise for simulating complex computer attacks in the context of IDS testing that combines vividness with the possibility of describing both deterministic and stochastic system is the use of a Petri-net apparatus [14].

The authors of [15] suggested that one could simulate computer attacks using modified Petri nets, which are generalized stochastic Petri nets with delays of some kind, restricting arcs, and weighted transitions that are necessary for the adequate description of network computer attacks.

The software implementation of such models makes it possible to synthesize the stochastically varied network traffic of complex computer attacks. Thus, the conditions for the development of a statistically consistent testing technique that makes it possible to evaluate the efficiency of IDSs for the detection of complex multistage computer attacks are provided.

## CONCLUSIONS

The known techniques for the testing and certification of network intrusion detection systems are focused, first of all, on the evaluation of IDS efficiency and functionality. The mode of a network attack that is accepted in these techniques is reduced to a set of malicious software modules that are periodically launched by an operator (or special software). As a result, the techniques are free from procedures that allow the evaluation of the potentialities of the tested systems for the detection and identification of thoroughly planned multistage computer attacks as "new" attacks whose signatures are absent in the IDS database.

The authors of this paper believe that simulation of complex computer attacks in order to synthesize the stochastically varied network traffic of the attacking action is the most promising method for ID testing. The obtained network traffic can be used many times for the statistically consistent testing of different intrusion detection systems. Moreover, the structure of the software–hardware test bed on which the testing is performed can be drastically simplified (in the extreme case, to one computer with several network adapters and special software for reproducing the traffic of attacks.

## REFERENCES

1. Puketza, N.J., Zhang, K., Chung, M., et al., A Methodology for Testing Intrusion Detection Systems, *IEEE Trans. Soft. Eng.*, 1996, vol. 22, no. 10, p. 719.
2. Lippmann, R.P., et al., Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation, *Proc. DARPA Inform. Surv. Conf. Exp. (DISCEX)*, 2000, vol. 2, pp. 12–26.
3. Kendall, K., *A Database of Computer Attacks for the Evaluation on Intrusion Detection Systems*, Cambridge: Massachusetts Inst. Technol., 1999.
4. Weber, D., *A Taxonomy of Computer Intrusions*, Cambridge: Massachusetts Inst. Technol., 1998.
5. McHugh, J., Testing Intrusion Detection Systems: a Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory *ACM Trans. Inform. System Sec. (TISSEC)*, 2000, vol. 3, no. 4, pp. 262–294.
6. van Trees, H.L., *Detection, Estimation and Modulation Theory. Part 1: Detection, Estimation and Linear Modulation Theory*, Tikhonov, V.I., Ed., New York: Wiley, 1968; Moscow: Sov. Radio, 1972.
7. Ulvila, J.W. and Gaffney, J.E., Evaluation of Intrusion Detection Systems, *J. Res. Nat. Inst. Stand. Technol.*, 2003, vol. 108, no. 6, p. 453.
8. Lippmann, R.P., et al., The 1999 DARPA Off-Line Intrusion Detection Evaluation, *Computer Networks: The Int. J. Comp. Telecom. Networking*, 2000, vol. 34, no. 4, pp. 579–595.
9. Durst, R., Champion, T., Witten, et al., Testing and Evaluating Computer Intrusion Detection Systems, *Comm. ACM*, 1999, vol. 42, no. 7, pp. 53–61.
10. Massicotte, F., Gagnon, F., Labiche, et al., Automatic Evaluation of Intrusion Detection Systems, *Proc. 22nd Annual Computer Security Applications Conf. (ACSAC)*, 2006, pp. 361–370.
11. Vigna, G., Robertson, W., and Balzarotti, D., Testing Network-Based Intrusion Detection Signatures Using Mutant Exploits, *Proc. 11th ACM Conf. on Computer and Communications Security*, 2004, pp. 21–30.
12. Neohapsis Labs. http://osec.neohapsis.com. (Accessed May 25, 2008).
13. NSS Labs. http://nsslabs.com/certification/ips/NIPS%20Methodology_v5_20.pdf. (Accessed May 25, 2008).
14. Peterson, J.L., *Petri Net Theory and the Modeling of Systems*, New York: Prentice-Hall, 1981; Moscow: Mir, 1984.
15. Khor'kov, D.A., About Possibility of Use of Petri Net Mathematical Apparatus for Computer Attack Modeling, *Dokl. Tomsk. Gos. Univ. Sistem Upravl. Radioelektr.*, 2009, No. 1(19), part 2, pp. 49–50.