Generation of Source Data for Experiments with Network Attack Detection Software

Igor Kotenko[.1,a], **Andrey Chechulin**[.2.,b] and **Alexander Branitskiy** [3.,c]
[1]Head of the Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). St.Petersburg, Russia.
[2]Senior researcher, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). St.Petersburg, Russia.
[3]Junior Researcher, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). St.Petersburg, Russia.

E-mail: [a] ivkote@comsec.spb.ru, [b] chechulin@comsec.spb.ru,
[c] branitskiy@comsec.spb.ru

Abstract. The paper suggests a new approach for traffic generation and the architecture of the software tool for evaluation of attack detection and response mechanisms. To assess the proposed approach the automatic network attack detection and response mechanism Threshold Random Walk (TRW) was chosen and implemented. The results of evaluation of this mechanism by the proposed software tool are presented.

1. Introduction

Currently generation of network traffic is an important issue when testing the distributed systems and conducting the research related to the detection of network attacks, including forecasting of anomalous activity in enterprise networks and explanation of self-similarity properties of network traffic.

The complexity of such problems is due to a clear lack of open data sets that are suitable for the tasks of traffic studies and real application. Moreover, existing data sets usually contain a limited sample of network attack classes of interest. In these circumstances, it becomes necessary to generate such network traffic, that is close to the real and contains a number of attacks of current interest.

There are several ways to generate the network traffic for experiments. The first (W1) is to use the real networks for traffic collection. This option gives the closest to real world examples, but most likely this traffic will not contain attacks or will contain very small amount and variety of attacks. The second (W2) is to use a honeynet for traffic collection. It is not always possible to create such network and to attract malefactors. The third (W3) is to use a network simulator. This approach can be very effective, but it is hard in installation and tuning. The fourth (W4) is to use existing traffic dumps. This is the most common way, but the traffic will be static, so there will be no possibility to change the environment. The fifth (W5) consists in usage of network traffic generators. Such generators usually can provide traffic with a lot of preferences, but the result can be very far from real world examples. Finally (W6) it is possible to use combination of above mentioned techniques. This option can combine the advantages of several ways and will be described in this paper.

The ideal source for experiments related to the analysis of network traffic and detecting the attacks is a live traffic, which is obtained as a result of the recording of network packets circulating in a large corporate system. However, since such a possibility is not always available, one possible solution to this problem is the use of network traffic generators.

The designed generator of network traffic must satisfy a number of requirements: (R1) it should take into account all the features of any necessary network application and typical attacks for it; (R2) it should be minimized to the presence of any network traffic characteristics resulting from its artificial generation; (R3) an automatic generation mechanism of network traffic for a given configuration of network equipment and the whole environment in general should be provided; (R4) a small amount of resources should be consumed when generating traffic; (R5) the labels of anomalous packets or connections should be provided.

The ways to generate the traffic and the considered requirements are summarized in Table 1. The symbol '+' indicated that the requirement Ri presented in the column *i* is satisfied by the way for generating the traffic Wj located in the row *j*. Otherwise, the requirement is not satisfied.

Table 1. Ways to generate traffic and considered requirements.

	R1	R2	R3	R4	R5
W1	+	+	–	+	–
W2	–	+	–	+	+
W3	+	+	–	–	–
W4	–	+	–	+	+
W5	–	–	+	+	+
W6	+	+/–	+	+	+

It should be noticed, that the traffic can be divided in two groups: the mixed set of packets and the cohesive packet sequences. The peculiarity of the former is the simplicity of operation and manual control over the contents of each packet, while the feature of the second is the ability to create the logical connections using a variety of underlying protocols. This sequences are characterized, for example, by the number of sent packets, a time between packets, the models of user behaviour, etc.

The main contribution of the paper lies in the fact that it offers a new approach for evaluation of network attack detection and response mechanisms that works with the traffic. This approach allows us to combine advantages of the real world traffic (traces) with possibilities of tuning of malware activities. The paper presents the approach for traffic generation and the architecture of the software tool to evaluate different attack detection and response mechanisms. To evaluate the proposed approach the automatic network attack detection and response mechanism Threshold Random Walk (TRW) was chosen and implemented. The results of evaluation by the proposed software tool are presented. The organization of the paper is as follows. Second section provides the analysis of existing approaches for traffic generation or collection. In third section we describe the developed testbed for experiments based on the proposed approach for traffic generation. Fourth section considers the mechanism for attack detection and response based on traffic analysis. In fifth section the evaluation of the attack detection and response mechanism by the developed testbed is presented. In sixth section the conclusions and future research directions are discussed.

2. Related works

There are several network traffic data sets (dumps) with attacks traces. We consider five examples (MAWI, DARPA, CAIDA, DRDC and CDX) the most popular for research in cyber security.

Data Set MAWI represents files with images of real connections transmitted through the backbone Pacific ocean backbone between Japan and the United States [1]. To ensure the confidentiality of transmitted data, the network packets are captured using tcpdump analyzer and truncated so that they contain only headers. A few different types of attacks were presented, the overwhelming majority of which includes port scanning and DoS attacks. For marking the connections four detectors are used

[2], and the severity level of the classified connection depends on the number of detectors, which recognize the connection as anomalous.

The data set DARPA of Linkoln Laboratory [3] contains network traces obtained by simulating the network interaction between hosts within one of the DARPA projects. In 1998 such data were collected for 7 weeks with a total volume including log files greater than 7 GB. In 1999 data were collected for 5 weeks (total volume of about 10 GB). This set is one of the most popular for scientific research, but it has a significant drawback – it is outdated.

The data set CAIDA [4] contains one hour traffic which includes DDoS attacks with total volume 21 GB. This set represents only anomalous traffic from the attacker and the corresponding responses from the victim. Traffic that does not contain attacks is removed as much as possible.

The data set DRDC (Defence Research and Development Canada) [5] is composed of more than 7000 documented attack scenarios and provides network traffic in pcap format, parameters in XML format and the output of commands in text form. A total of 94 exploitability programs are presented, there are scripts and program sources for the implementation of the attacks. The data set comprises 49 different vulnerabilities using 17 network services, 108 different target operating systems. A significant drawback of this data set that it is not publicly available.

The data set CDX (Cyber Defense Exercise) [6] includes 23 pcap-files generated in April 2009 as the result of the competition for building the secure network environments and protecting them from the network attacks. Its special feature is the presence of a set of log files obtained as a result of functioning the server applications (Web and DNS) and IDS Snort.

Network traffic generators can be implemented as software or hardware components. Using these generators one can set various network traffic parameters, including traffic direction (addresses of a source and a receiver), packet length, protocol type, packet rate, etc.

As it was mentioned before the network traffic generators can be divided into two categories: generators on the packet level and generators based on cohesive packet sequences.

One of the most famous examples of the first class utilities is *hping3*. Some of the popular tools of the second class include *ostinato*, which supports many different protocols from the channel to application layer of the OSI model. Since the generators of the second class are of the greatest interest, we will consider briefly the corresponding relevant work concerning them.

Harpoon [7] is a research prototype generating the network traffic on TCP and UDP level. The features of this software tool is the ability to generate such traffic that is largely statistically similar to the real traffic. The downside is the lack of opportunity to create network flows that would be able to simulate the data transmission between the network applications (for example, between the Internet browser and the Web-server). This system has a three-layer architecture, represented by the file level, session level and user level.

Swing [8] is without drawbacks of Harpoon and can take into account the properties of packets, depending on the highest level protocol. Swing's architecture considers different features of the network traffic, typical users, sessions, connections and network characteristics.

In [9] the fast algorithm of fractional Gaussian noise was developed for improving the quality characteristics of the network traffic and approximating the parameters of self-similar stochastic processes of the network traffic.

After analyzing the existing data sets and traffic generators, it can be concluded that the existing solutions do not satisfy all the requirements we considered and are focused on the generation of certain classes of data according to the protocol type, application, etc. Since the simultaneous and complete fulfilment of these conditions is often unfeasible in practice with the help of such generators, we propose to develop a combined approach, which allows to take into account, if not all, then at least the maximum number of these requirements.

This approach is based on combining different sources of traffic, including existing data sets (real traffic traces) as well as output of normal and (or) attack traffic generators. Such approach is intended to combine advantages of the real world traffic (traces) with possibilities of tuning of traffic of normal applications and malware activities.

3. Proposed approach

The combined approach consists in *application of various scenarios of traffic traces with the insertion of a needed traffic*. The needed traffic in this case is the attack traffic and (or) the traffic of some applications, for example, “fast” applications, such as NetBIOS/NS or P2P. |It is proposed to combine both the real traffic traces and the generated traffic (plugging various modules of traffic models).

The proposed common architecture of the tool for generation and analysis of the legitimate and network attack traffic is in Figure 1.

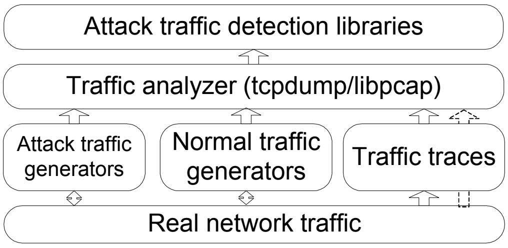

Attack traffic detection libraries
Traffic analyzer (tcpdump/libpcap)
Attack traffic generators
Normal traffic generators
Traffic traces
Real network traffic

Figure 1. Traffic generation and analysis tool architecture.

The tool is based on using the real network traffic. This traffic can be used to create traffic traces, extract parameters for legitimate and attack traffic generators and their tuning. Traffic traces can be both downloaded from the corresponding repositories and recorded due to tcpdump/libpcap. Libpcap is the application program interface for packet capturing from the network interface. Tcpdump is intended for traffic recording due to libpcap. Traffic analyzer is based upon tcpdump/libpcap and allows receiving and synchronizing the traffic from various sources.

The main idea of evaluation methodology is to run a series of experiments using the software tool for various values of input parameters measuring the defense parameters. It is supposed that measuring the effectiveness and efficiency parameters of automatic detection and response techniques and their combination as well as fulfilling their analysis enables to select the rational (or optimum) parameters of functioning and (or) developing the dynamic strategies of operation.

To simulate the impact of the defense methods the traffic filter (if defense mechanism blocks some traffic than it will not reach the detection mechanisms) is added to the traffic generator (Figure 2) [10].

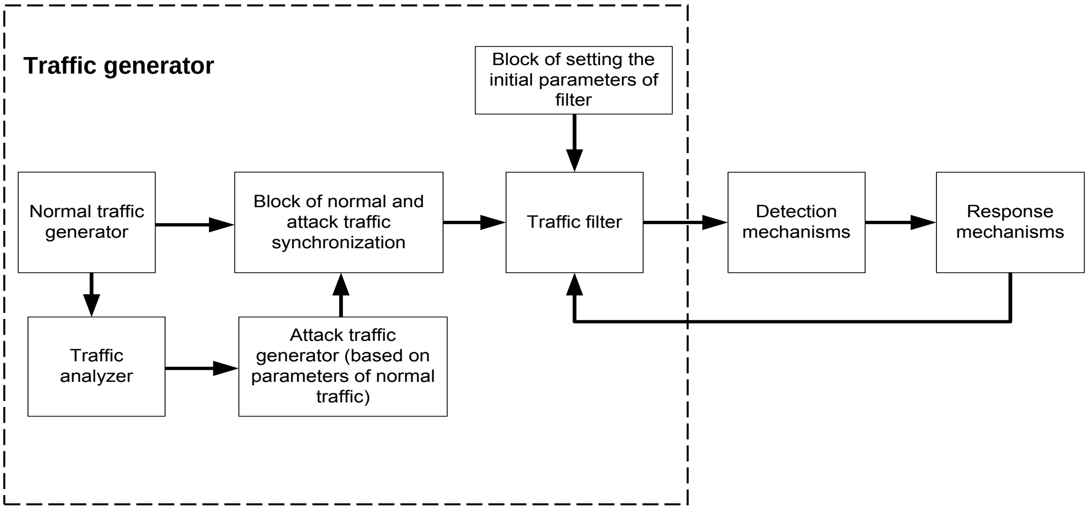

Traffic generator
Block of setting the initial parameters of filter
Normal traffic generator
Block of normal and attack traffic synchronization
Traffic filter
Detection mechanisms
Response mechanisms
Traffic analyzer
Attack traffic generator (based on parameters of normal traffic)

Figure 2. General scheme of traffic generator and its interrelation with defense mechanisms [10].

The normal traffic generator implements reading traffic traces from files. *The block of normal and attack traffic synchronization* serves for transferring the network packets to defense mechanisms in the time ordered sequence under simultaneous use of several traffic sources. *The traffic analyzer* calculates traffic parameters. These are parameters are used for attack traffic generation, and also to investigate the correspondence of traffic parameters and the defense mechanisms which work in the best way at the given traffic. *The attack traffic generator* allows generating the traffic for both already known attacks, and unknown attacks which can appear in the future. *The traffic filter* – as traffic traces have answers to the requests blocked by response methods, the traffic filter is used to simulate dynamics in traffic records. *The block of setting the initial parameters of the filter* allows using access control list (ACL) to ignore connections to certain addresses, ports and protocols. *Mechanisms of detection and response* are the defense mechanisms under investigation. They make decisions on presence of attacks on hosts and protecting against them.

Traffic is supposed to be preprocessed to raise the defense effectiveness. The preprocessing is in ACL filtering, “dark addresses” determination and processing the honeypot messages, etc. ACL is assigned with the traffic filtering rules, and traffic is filtered before it comes to defense mechanism. The rules are defined with addresses and ports of such hosts that have legitimate applications installed creating a lot of connections.

If traffic source is represented by the real traffic, then the real time interval of the experiment is set. In case of the recorded traffic, the time interval inside the recorded traffic is set.

Let us consider the one of the attack types − network scanning. *TCP scanning* is performed by initiating a TCP connection. Scanning tool creates a thread and sends TCP-SYN packet to the victim host. If it will not reply, the packet will be resent after 3 and 6 seconds accordingly. The scanning stops after 3 unsuccessful tries. Scanning speed depends on the amount of allowed threads and simultaneous connections on this host. It is defined by the operating system. *UDP scanners* usually send only one packet without retries. Scanning speed is limited by the channel bandwidth. Various scanning strategies are defined to choose the scanned hosts.

Table 2 represents parameters of different scanners.

Table 2. Parameters for scanners simulation

	Scanner 1	Scanner 2	Scanner 3
Protocol	TCP	UDP	UDP
Packet generation frequency (per second)	5.65	4000	357
Length (bytes)	40	404	796-307
Ports (source/destination)	any/80	any/1434	4000/random
Scanning strategy	1/8 of time the random IPs, 1/2 of time IPs from the same /8 subnet (X.*.*.*), 3/8 of time IPs from the same /16 subnet (X.Y.*.*)	the source formula (to receive the next pseudo-random address from the previous) is as follows: x' = (x * 214013 + 2531011) mod 2^32	Random distribution

There are more parameters that can be specified for scanner: connection type - connections can be either TCP-based or UDP-based; frequency of packet generation (number of packets (connections) per second); scan speed variation - the speed at which the scanner scans can be constant or varied; scan type (destination address and port choice technique) - random-scanning, sequential-scanning, permutation-scanning, partition-scanning, local-preference-scanning, topological-scanning, hitlist-scanning, combination; probability of successful TCP connection; packet size; probability of RST packet receiving as response to TCP connection request; complete or incomplete closing of TCP connection; probability of UDP packet receiving as response to sent UDP packet; address and port spoofing technique; number of addresses used; etc.

4. Scanning detection technique for experiments

The analysis of different techniques for attack traffic detection was presented in previous authors' papers [10-12]. In our research we investigated several techniques: virus throttling (VT) mechanisms, mechanisms based on failed connections (FC) analysis, mechanisms based on Threshold Random Walk (TRW), credit based (CB) mechanisms, DNS-based mechanisms; and combined detection and response mechanisms. In this paper we describe only basic TRW technique for attack detection and response just to show how it can be evaluated by the proposed approach.

Threshold Random Walk (TRW) technique [13] uses a random walk to decide whether a new connection initiated by a host is benign or malicious. It keeps a ratio for each host and in the case of a successful connection started by that host, increases its ratio by some value, making the ratio farther from a fixed threshold (and vice versa in the case of failed connection attempt).

The following supposition is offered to discover hosts which took part in the attacks (here it is the scanning attack). This method uses TCP protocol. TCP SYN, TCP RST, TCP SYN ASK packets and fields “source IP” and “destination IP” are analyzed.

The method of sequential hypothesis testing is used to analyze the host that shows high network activity and might be source of scanning:

- Let H_1 be a hypothesis that the host r shows high network activity (it is the source of scanning), H_0 - a hypothesis that the host does not show the high network activity (it isn't the source of scanning). Let Y_i be a variable describing the connection attempt to the *i*-th host. This variable can have the following values:

$$Y_i = \begin{cases} 0, \text{ if connection is established} \\ 1, \text{ if connection is not established} \end{cases}.$$

- Let us now assume that conditional on the hypothesis H_j the random variables $Y_i \mid h_j = 1,2,...$ are independent and identically distributed. Then we can express the distribution of the Bernoulli random variable Y_i as:

$$\begin{aligned} P_r[Y_i = 0 \mid H_0] = \theta_0, \quad P_r[Y_i = 1 \mid H_0] = 1 - \theta_0 \\ P_r[Y_i = 0 \mid H_1] = \theta_1, \quad P_r[Y_i = 1 \mid H_1] = 1 - \theta_1 \end{aligned}. \tag{1}$$

- The observation that a connection attempt is more likely to be a success from a benign source than a malicious one implies the condition: $\theta_0 > \theta_1$.
- According to these two hypotheses and the real state four possible decisions can be selected (Table 3):

Table 3. Possible decisions

	H_1 is selected	H_0 is selected
There is an attack in the traffic	Scanning detection	False negative
There is no attacks in the traffic	False positive	No attack

- Let us evaluate the performance of detection by two variables: detection probability P_D and false positive probability P_F. So the algorithm has appropriate quality if the following conditions are correct:

$$P_D \geq \beta \text{ and } P_F \leq \alpha, \ \alpha\text{=0.01 } \beta\text{=0.99}. \tag{2}$$

- As each event is observed we calculate the likelihood ratio:

$$\Lambda(Y) \equiv \frac{\Pr[Y \mid H_1]}{\Pr[Y \mid H_0]} = \prod\nolimits_{i=1}^{n} \frac{\Pr[Y_i \mid H_1]}{\Pr[Y_i \mid H_0]}, \tag{3}$$

where Y is the vector of events that was analyzed and $\Pr[Y_i \mid H_i]$ is the conditional probability mass function of the event stream Y given that model H_i is true; and where the second equality in (3) follows from the independent and identically-distributed random variables assumption.

- Then Λ is compared to an *upper* threshold η_1 and a *lower* threshold η_0. The results of comparison show the presence or absence of an attack actions on the host:

$$\Lambda(Y) \le \eta_0 \Rightarrow H_0\,, \quad \Lambda(Y) \ge \eta_1 \Rightarrow H_1.$$

If $\eta_0 < \Lambda(Y) < \eta_1$ then we should wait till the next observation and update $\Lambda(Y)$.

In [13] it is shown that

$$\eta_1 = \frac{\beta}{\alpha}, \quad \eta_0 = \frac{1-\beta}{1-\alpha}.$$

The proposed method of sequential testing hypotheses can be represented graphically as follows (Figure 3).

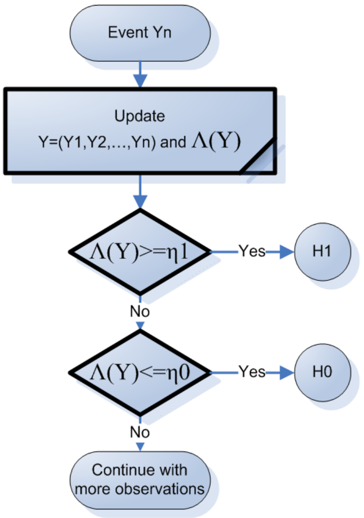


Figure 3. Sequential Hypothesis Testing scheme [13].

Input parameters of TRW algorithm: source IP-address; destination IP-address.

Control parameters of TRW algorithm: Previous Contacted Hosts (PCH) list for each host; First Contact Connection (FCC) queue.

For each host *s* TRW maintains three variables: D_s - is the set of distinct IP addresses to which *s* has previously made connections; S_s - reflects the decision state (PENDING, H_0, H_1); L_s - likelihood rate.

The following steps are executed for each record:

1. The sequence is not analyzed, if S_s is not PENDING.

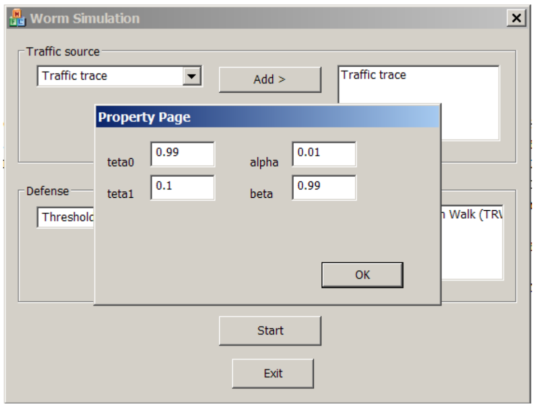
2. Determine whether the connection is successful or not (a connection is successful if it elicited the SYN ASK response).

3. Check whether the address d is already in D_s. If so, skip the further processing and go to the next request.

4. Add d in D_s and update L_s using (3).

5. If $L_s \geq \eta_1$ set host status to H_1. If $L_s \leq \eta_0$ set host status to H_0.

TRW parameters dialog is presented in Figure 4. When the parameters are set, user presses the "OK" button and thus finishes defense method customization. In the main form, when traffic source is also selected, the "Start" button for simulation start-up.



teta0 0.99
teta1 0.1
alpha 0.01
beta 0.99

Figure 4. TRW parameters dialog.

5. Experiments results

For reasoned comparison of defense mechanisms it is necessary to select optimal parameters of launching of every one of them. Selection of parameters was performed on two types of traffics, recorded in files, with artificially generated attack traffic, admixed to them.

For selection of the best defense mechanisms parameters, two traffic types were used: (1) enterprise-level network traffic with big set of different applications (including vulnerability scanners) without prevalence of any traffic; (2) local area network traffic with prevalence of P2P applications.

The specified traffics were mixed with attack traffic.

Selection of such traffic types is reasoned by the fact that the mechanisms of detection and response against network attacks should precisely detect them both in traffic of legitimate applications that are used for networks purposes, and in traffic of applications that create great number of connections with different nodes. The latter is especially important, because traffic of applications that create great number of connections with different nodes resembles the behavior of network attacks.

The following traffic records were used for parameters selection:

- *Record of traffic lbl-internal.20041216-1618.port008.dump.anon* (181 Mb) with scanner traffic lbl-internal.20041216-1618.port008.dump.anon-scanners (870 Kb). It is one of the traffic records from the library [14] that includes more than 100 hours of records of enterprise level network traffic for the period from October 2004 till January 2005. The specified record is one of the biggest in the library with corresponding to it scanners traffic. By different applications traffic content this record is typical for LBNL library. Exception makes a choice of a file for adjustment of the DNS-based mechanism: the traffic *lbl-internal.20050107-1323.port026.dump.anon* (163 Mb) with the traffic of the scanner *lbl-internal.20050107-1323.port026.dump.anon-scanners* (2,81 Mb) (which contains sufficient amounts of DNS-traffic) was investigated.
- *Record of traffic p2p_emule_web_upload_download* (101 Mb). It is one of records, specially created on our testbed, that include a number of traffic records for a local area network, in which traffic of P2P file exchange networks of different types prevails. The specified record is a traffic formed by a client of P2P network E-donkey called Emule. At the same time WWW traffic was also created in the network. This traffic resembles a attack traffic by its structure,

because this P2P client attempts and creates multiple connections with different nodes. Selection of this record is also associated with the fact that Emule is one of the most widely spread p2P clients, as well as by that WWW traffic is present in the record.

Three types of scanners were admixed to the specified traffics:

- Scanner 1 has the following parameters: generation time—during the whole traffic time recorded in the file (connTime=-1), protocol—TCP (protocol=TCP), connection completion type—full (endOfConnection=FULL), addresses scanning—random (scanType=RANDOM), frequency—50 packets a second (scanSpeed=50), packet size—500 bytes (packetSize=500), successful connection probability— 0.3 (connProb=0.3), TCP-RST receive probability— 0.3 (rstProb=0.3).
- Scanner 2 is characterized by the following parameters: generation time—during the whole traffic time recorded in the file (connTime=-1), protocol—TCP (protocol=TCP), connection completion type—full (endOfConnection=FULL), addresses scanning—random (scanType=RANDOM), frequency—6 packets a second (scanSpeed=6), packet size—50 Kbytes (packetSize=50*1024), successful connection probability— 0.2 (connProb=0.2), TCP-RST receive probability— 0.6 (rstProb=0.6).
- Scanner 3 has the following parameters: generation time—400 sec (connTime=400), protocol—UDP (protocol=UDP), addresses scanning—random (scanType=RANDOM), frequency—50 packets a second (scanSpeed=50), packet size—404 bytes (packetSize=404), successful connection probability— 0.2 (connProb=0.2).

Selection of the given scanners types is reasoned by the following. It was necessary to estimate work of mechanisms both for known scanners, using different protocols, and for potentially possible or future ones.

For each method one-two main parameters were selected, changing of which to the greatest extent effects the number of errors: permits the attack traffic and block the normal traffic. In the case of one parameter the search of local minimum of errors was performed, that is of such parameter value, shifts to the greater or to the lesser side from which lead to increase of errors. In the case of two parameters the search was performed iteratively on each of them.

The aim of experiments with TRW was to find the optimal values for θ_0 and α. There were performed more than 500 experiments, and the experiments showed that the optimal parameters values for θ_0 and α in TRW mechanism are: Normal traffic, mixed with scanner 1, 2 or 3 traffic — 0.52 and 0.001 respectively; Peer-To-Peer traffic, mixed with scanner 1 traffic — 0.5005 and 0.001 respectively; Peer-To-Peer traffic, mixed with scanner 2 traffic — 0.502 and 0.0011 respectively; Normal traffic, mixed with scanner 3 traffic — 0.5005 and 0.01 respectively.

The suggested approach for traffic generation allowed us to provide controlled series of experiments and select the optimal parameters for usage of the TRW mechanism. We experimented also with other defense mechanisms (for example, VT, FC, CB, DNS-based and combined mechanisms) and also selected the optimal parameters. Thus , we validated the suggested approach and demonstrated that it can be implemented for design, adjustment and evaluation of different network attack detection and response mechanisms.

6. Conclusion

The main results of the research presented in the paper are as follows. The challenge was formulated, and the main requirements for network traffic generation for evaluation of detection and response techniques against network attacks were developed. The current status of network traffic generation was considered. The new approach to evaluate the mechanisms for network attack detection and response was proposed. It includes the technique for simulation of network attacks by mixed traffic generation. The proposed simulation tool architecture was presented. This tools allows to evaluate the main parameters of detection and response techniques. The series of experiments (using variety of scenarios) for different values of input parameters was conducted. The evaluation of automatic detection and response mechanism was performed by the software which implements the proposed approach. Suggestions on application and further improvement of the proposed approach to assess automatic detection and response mechanisms against network attacks were developed.

The future work will be devoted to the development of the software benchmarking system for evaluation of attack detection and response mechanisms on the basis of the proposed approach for traffic generation. It is suggested to create the set of various traffic sources (creation and use of already generated traffic traces, generators of malicious and legitimate traffics) that will cover all aspects of network functioning in normal mode and attack actions. Each benchmark could consist of testing the mechanism on various traffic combinations, in various modes and with various parameters.

Acknowledgments

The work is performed by the grant of RSF #15-11-30029 in SPIIRAS.

References

1. Fontugne R, Borgnat P, Abry P and Fukuda K 2010 MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking *Proc. Int. Conf. on emerging Networking EXperiments and Technologies* (Philadelphia, USA) pp 1–12 http://conferences.sigcomm.org/co-next/2010/CoNEXT_papers/08-Fontugne.pdf
2. MAWILab http://www.fukuda-lab.org/mawilab/documentation.html
3. DARPA Intrusion Detection Data Sets https://www.ll.mit.edu/ideval/data/
4. The CAIDA "DDoS Attack 2007" Dataset http://www.caida.org/data/passive/ddos-20070804_dataset.xml
5. McKenna J and Treurniet J Network Attack Reference Data Set 2004 *Technical Memorandum* (Defense R&D, Canada, Ottawa) http://cradpdf.drdc-rddc.gc.ca/PDFS/unc88/p523954.pdf
6. The United States Military Academy Data Sets http://www.usma.edu/crc/SitePages/DataSets.aspx
7. Sommers J, Kim H, and Barford P 2004 Harpoon: A Flow-Level Traffic Generator for Router and Network Tests *Proc. of the Joint Int. Conf. on Measurement and Modeling of Computer Systems* vol 32 (New York, NY, USA, ACM) pp 392–392.
8. Vishwanath K and Vahdat A 2016 Realistic and Responsive Network Traffic Generation *Proc. of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (New York, NY, USA, ACM) pp 111–122.
9. Paxson V 1997 Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic *Proc. of the 1997 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (New York, NY, USA, ACM) pp 5–18.
10. Kotenko I 2009 Framework for Integrated Proactive Network Worm Detection and Response *Proc. of the 17th Euromicro International Conference on Parallel, Distributed and network-based Processing* (Los Alamitos, CA, USA IEEE) pp 379-386.
11. Kotenko I V, Chechulin A A, Doynikova E V 2011 Combining of Scanning Protection Mechanisms in GIS and Corporate Information Systems *Proc. of the 5th International Workshop on Information Fusion and Geographical Information Systems* vol 5 (Berlin Heidelberg Springer) pp 45–58.
12. Branitskiy A A, Kotenko I V 2015 Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers *Proc. of the 18th IEEE International Conference on Computational Science and Engineering* (Los Alamitos, CA, USA IEEE) pp 152–159.
13. Jung J, Paxson V, Berger A W and Balakrishnan H 2004 Fast portscan detection using sequential hypothesis testing *Proc. of the 2004 IEEE Symposium on Security and Privacy* (Los Alamitos, CA, USA IEEE) pp 211-225.
14. LBNL/ICSI Enterprise Tracing Project http://www.icir.org/enterprise-tracing/index.html