Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2024. Т. 24, вып. 3. С. 452–462
Izvestiya of Saratov University. Mathematics. Mechanics. Informatics, 2024, vol. 24, iss. 3, pp. 452–462
https://mmi.sgu.ru https://doi.org/10.18500/1816-9791-2024-24-3-452-462, EDN: OSEMWU

Article

Detection of sources of network attacks based on the data sampling

E. S. Sagatov, A. M. Sukhov✉, V. V. Azhmyakov

Sevastopol State University, 33 Universitetskaya St., Sevastopol 299053, Russia

Evgeny S. Sagatov, sagatov@ya.ru, https://orcid.org/0000-0001-9780-8302, AuthorID: 819717
Andrei M. Sukhov, AMSuhov@sevsu.ru, https://orcid.org/0000-0001-6948-4988, AuthorID: 158417
Vadim V. Azhmyakov, vvazhmyakov@mail.sevsu.ru, https://orcid.org/0000-0003-3634-6786

Abstract. This article defines the rules for finding the threshold values for the main network variables used to detect network intrusions under conditions of limited data sampling. The sFlow technology operates with a limited sample of packets, and one packet out of 50 can be analyzed, but this value can reach 5000. The main conclusion is that the product of the threshold value and sample resolution remains a constant value. The article defines the size of the maximum resolution, at which an attack with a given threshold can be detected. Based on the experimental data, this hypothesis was tested; considering the experimental error, it was verified.

Keywords: thresholds for detecting DDoS attacks, sFlow data sampling, rank distributions in network security
Acknowledgements: The authors acknowledge Sevastopol State University (SevSU) for the Research Grant 42-01-09/253/2022-1.
For citation: Sagatov E. S., Sukhov A. M., Azhmyakov V. V. Detection of sources of network attacks based on the data sampling. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2024, vol. 24, iss. 3, pp. 452–462. https://doi.org/10.18500/1816-9791-2024-24-3-452-462, EDN: OSEMWU
This is an open access article distributed under the terms of Creative Commons Attribution 4.0 International License (CC-BY 4.0)

Научная статья
УДК 004.7

Обнаружение источников сетевых атак на основе выборки данных

Е. С. Сагатов, А. М. Сухов✉, В. В. Ажмяков

Севастопольский государственный университет, Россия, 299053, г. Севастополь, ул. Университетская, д. 33

Сагатов Евгений Собирович, кандидат технических наук, доцент, научный сотрудник научно-исследовательской лаборатории «Прикладная математика и суперкомпьютерные вычисления», sagatov@ya.ru, https://orcid.org/0000-0001-9780-8302, AuthorID: 819717
Сухов Андрей Михайлович, доктор технических наук, ведущий научный сотрудник научно-исследовательской лаборатории «Прикладная математика и суперкомпьютерные вычисления», AMSuhov@sevsu.ru, https://orcid.org/0000-0001-6948-4988, AuthorID: 158417
Ажмяков Вадим Викторович, Dr. rer. nat. habil., профессор, заведующий научно-исследовательской лабораторией «Прикладная математика и суперкомпьютерные вычисления», vvazhmyakov@mail.sevsu.ru, https://orcid.org/0000-0003-3634-6786

© *Sagatov E. S., Sukhov A. M., Azhmyakov V. V., 2024*

Аннотация. В статье определены правила нахождения пороговых значений для основных сетевых переменных, используемых для обнаружения сетевых вторжений в условиях ограниченной выборки данных. Технология sFlow оперирует с ограниченной выборкой пакетов, причем анализироваться может один пакет из 50, но это значение может доходить и до 5000. Наш основной вывод состоит в том, что произведение порогового значения и разрешения выборки остается постоянной величиной. В работе определен размер максимального разрешения, при котором атака с заданным порогом может быть обнаружена. На основании собранных во время эксперимента данных было проведено тестирование данной гипотезы. С учетом ошибки эксперимента эта гипотеза подтверждается.

Ключевые слова: пороговые значения для распознания DDoS атак, выборка данных sFlow, ранговые распределения в сетевой безопасности

Благодарности: Авторы выражают благодарность Севастопольскому государственному университету за поддержку в рамках проекта 42-01-09/253/2022-1.

Для цитирования: *Sagatov E. S., Sukhov A. M., Azhmyakov V. V.* Detection of sources of network attacks based on the data sampling [*Сагатов Е. С., Сухов А. М., Ажмяков В. В.* Обнаружение источников сетевых атак на основе выборки данных] // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2024. Т. 24, вып. 3. С. 452–462. https://doi.org/10.18500/1816-9791-2024-24-3-452-462, EDN: OSEMWU

Статья опубликована на условиях лицензии Creative Commons Attribution 4.0 International (CC-BY 4.0)

Introduction

Despite the efforts made by law enforcement authorities and government regulators, online incidents are still numerous, the losses that bring destructive actions are growing, and the situation is similar in highly developed countries all over the world. Reliable protection of the telecommunications structure and the data stored is the most important task for government agencies and businesses. The fight against cybercrime makes even ideological opponents (like Russia and the United States) enter into talks and agree on cooperation.

In the present work, an effort is made to study the threshold method for detecting the sources of network attacks based on the data sampling. When conducting network attacks, a number of network variables take values that are many times higher than their average ones [1]. The developed approach involves finding the threshold values for the main network variables. If these values are exceeded, it is possible to ascertain the beginning of a network attack.

The creation of a full-fledged network monitoring structure for intrusion prevention is rather complicated, let alone too expensive. This is primarily due to the fact that the monitoring system should ideally intercept and analyze every data packet, which greatly increases the requirements for the computing power of the equipment. Such technologies from the very beginning tried to reduce the amount of information processed; the first technology for collecting information about NetFlow traffic was offered by Cisco in 1996 [2]. This technology involves storing information about the flows, which greatly reduces the amount of information; all the packets are processed, but the information is stored only about the connections (flows). Such a technology is oft-used in intrusion protection systems.

Nevertheless, in the context of the exponential growth of traffic in the global network, the equipment with NetFlow technology support is becoming more expensive, therefore, not everyone can afford it. Thus, a new technology for monitoring sFlow traffic, whose distinguishing feature is a selective analysis of traffic at the packet level, appeared [3]. This technology can analyze one network packet out of tens, hundreds, and even thousands, which is set by the administrator when the monitoring system is configured.

However, this way of monitoring, based on limited data sampling, raises a number of questions about intrusion detection technologies, including the issue of finding the new threshold values for network variables. This study implies the adaptation of intrusion detection and prevention technologies for a new method of traffic monitoring, involving a selective packet analysis method.

The novelty of the research proposed is stipulated by the usage of a unified mathematical

approach. This approach uses rank distributions for the statistical analysis. A number of important network variables, generated by an external single IP address when accessing a given server or local network, are to be identified. These variables include the frequency of accessing the web server (on a given port), the number of active threads, the amount of incoming TCP, UDP, and ICMP traffic, etc.

The experimental infrastructure allows measuring the values for the above network variables. The rank distributions are built based on sFlow data; the rank distribution curves are expected to lie lower (compared to when NetFlow data), depending on the sample resolution. The found threshold values are to be experimentally tested while conducting network attacks using the most common utilities widely applicable by attackers.

The article further includes an overview of attack detection techniques and the thresholding method (Section 1); consideration of rank distributions and recognition of attacks and their sources (Section 2); experiment to test the hypothesis about the threshold value (Section 3); discussion of the results obtained (Section 4); conclusions.

1. Related works

It has been more than 20 years since Cisco NetFlow was patented; extensive research has been conducted, and many applications have been developed. In the review [4], the authors considered the current development in the research in this area, highlighting the main perspectives and methodologies. The analysis showed that network security is a big part of such studies.

One of the first works that proposed a threshold approach to recognizing DDoS attacks was the work presented at the Defense Advanced Research Projects Agency (DARPA) conference [5]. Since 2015, there has been a surge of interest in network attack detection models using threshold-based algorithms. As stated in work [6], denial of service attacks (DoS) and distributed denial of service attacks (DDoS) are becoming more and more frequent violations of the global Internet; so, the authors proposed to improve the detection of distributed denial of service attacks based on the fast entropy method using the streaming analysis.

The same authors continued their research in work [7]. This article proposes an efficient statistical approach for attack detection based on the traffic characteristics and dynamic thresholding algorithm (the latter is used because both network activity and user behavior can change over time).

The increased rate of legitimate traffic flow and its similarity to the traffic flow during the attack made the DDoS problem even more urgent. In work [8], it is proposed a distributed intrusion detection system T-CAD, which calculates the normalized router entropy and compares it with various thresholds to effectively distinguish between legitimate traffic, DDoS attacks, and flash events. In the work [9], there is a review of the best-known anomaly-based intrusion detection methods.

Most approaches, which are based on the training with the use of neural networks, have a number of significant drawbacks. The main disadvantage is that when a new attack method appears, it will take several hours or even days before the qualification signs of a new type of attack are found; this time will be spent on collecting statistics and training.

In the field of network security, researchers have implemented various models to protect networks; in particular, Snort, the foremost open-source intrusion detection and prevention system, is one of them. Currently, intrusion detection system (IDS) is a growing technology in the field of network security, and many researchers contribute to its development using rule-based and anomaly-based methods. In work [10], the authors proposed a rule-based IDS with the new efficient port scan detection rules (EPSDR). These rules are used to detect real-time network naive port scanning attacks using Snort and the Basic Analysis Security Engine (BASE). BASE is used to view Snort results on a font web page because Snort doesn't have a graphical user interface (GUI).

As is known, DDoS attacks are a common threat to network security, and traditional

mitigation approaches have significant limitations in dealing with them. The work [11] reviews the main traditional approaches to DDoS, identifies and discusses their limitations, and proposes a software-defined network (SDN) model as a more flexible, efficient, and automated mitigation solution. This study focuses on the networks belonging to Internet Service Providers (ISPs) by the example of the implementation of SDN security in Verizon's networks.

The work [12] presents an analysis of the most relevant types of attacks based on the reflection and amplification of traffic. The methods, recommended for preventing such attacks, as well as the existing methods of protection, are given. The advantages and disadvantages of these methods are revealed. Further goals for the development of new methods of protection are set.

The SDN architecture has the potential to be used to modernize security and implement more effective threat countermeasures inherent in traditional data network architecture. In work [13], an overview of the SDN architecture and the OpenFlow protocol is given, an analysis of threats and technologies for their neutralization for the SDN architecture and the OpenFlow protocol are presented, and critical threats for those OpenFlow networks that may soon appear in the Russian Federation are identified; in addition, the ways to counteract these threats are proposed. According to Gartner (Technology Evolution Curve), SDN security technologies are at the peak of inflated expectations.

2. Rank distributions and recognition of anomalous network states

The analysis of network processes on the Internet quite often relies on non-standard types of distributions. In particular, many processes can be described using rank distributions. This type of distribution was first applied in the field of network technologies by Steve Glassman [14] in 1994. He was able to describe the process of reserving Internet traffic on proxy servers. Since then, the scope of rank distributions has gradually expanded to include the field of network security.

Rank distributions involve ordering the values of the p value under study in descending order. The sequence number of the value in the ordered list is called the rank i. With the help of rank distributions, it has been explained the existence of threshold values for many network variables, such as:

- the total number of active flows on the router of the protected network segment;
- the number of active flows that the external IP address generates;
- the number of packets that the external IP address generates;
- the amount of incoming TCP, UDP, and ICMP traffic from the external IP address;
- the number of requests for a fixed internet service (DNS, NTP, SNMP, etc.).

Currently, the works [15, 16] that describe the use of rank distributions for practical applications (including the field of network security) have been published.

As a rule, rank distributions are described by the Zipf law:

$$p_i = \frac{p_1}{i^\alpha}, \tag{1}$$

where p_1 is the largest value under study, i is the rank, and α is the indicator of the degree of distribution.

To detect an attack and identify its sources, two rank distributions are compared. The first of these distributions is constructed at the time when the network is in its normal state. The second distribution is built at the time of the attack. Previously, it was proposed to analyze the rank distributions for the number of flows that a single IP address generates [1]. It was determined that at the moment of attack, this value increases by at least an order of magnitude.

A graphical illustration of this attack detection method is shown in the diagram in Fig. 1.

The p^{tr} value can be used as a threshold. All the values on the upper curve above the p^{tr} threshold should be considered to be those reflecting the attack status of the network. Based on these points, it is easy to identify the attack IP addresses, the traffic from which should be blocked for a short period of time.

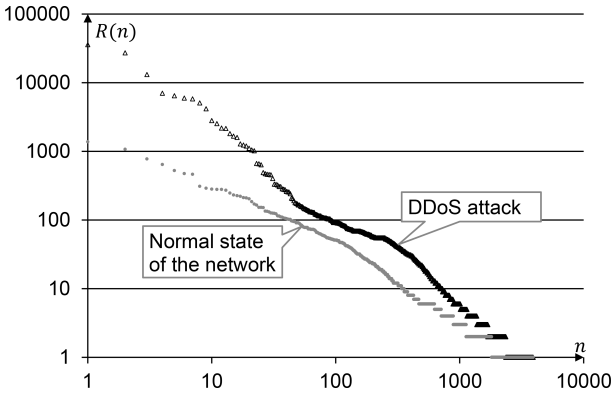

Fig. 1. Threshold finding

Note that the values along the axes in the diagram, shown in Fig. 1, are plotted in a logarithmic format. In this case, the Zipf distribution from (1) is a straight line

$$\lg p_i = \lg p_1 - \alpha \lg i. \tag{2}$$

The algorithm for finding the threshold value is described in detail in the work [1]. Since p^{tr} is calculated from NetFlow data, it can be assumed that the sampling resolution (for this value) is $N = 1$.

The present work intends to generalize the threshold value method for the case of sampled traffic analysis. In the introduction (recalling the problem statement), it was said that a complete analysis of all traffic is difficult due to the computational complexity of the problem and the large amount of resources required. The novelty is that not the entire incoming traffic can be analyzed but the limited sample of incoming traffic packets. Modern technologies involve the analysis of a certain sample of packets, and the sampling rate N can vary and reach 1 packet out of 5000 transmitted. In this case, the values of variables that exceed the threshold can also be confidently fixed. Only small flows that do not affect the definition of attack IP addresses are to be discarded. However, it is necessary to establish new threshold values p_k^{tr}, as well as find ways to measure them using the data from a limited sample with the resolution of N_k.

The task is to find how the threshold value p_k^{tr} will change, depending on the sampling rate N_k. To do this, the threshold values p_k^{tr} should be replaced with p^{tr}/N, since only one of N packets can be intercepted. Note again that the threshold value p^{tr} is fixed for packet sampling with $N_k = 1$.

In logarithmic coordinates, the equation of the straight line, describing the rank distribution for the sampling frequency N, is

$$\lg p_i = \lg p_1 - \lg N - \alpha \lg i. \tag{3}$$

This graph is parallel to the old line, but below it by $\lg N$.

Of particular interest in the sampled traffic analysis is the problem of determining the limiting sampling frequency N_lim, at which the threshold value p^tr can still be detected.

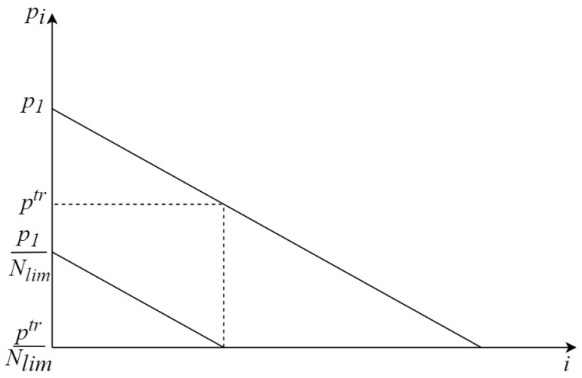

Fig. 2. Finding the limiting sampling frequency N_{lim}

Figure 2 shows the graph illustrating how the new threshold is calculated under conditions of limited data sampling. As shown earlier, the new curve, describing the rank distribution with the limiting sampling frequency N_{lim}, will be parallel to the old straight line but $\lg N_{lim}$ units lower.

In this case, the new threshold value p_{lim}^{tr}, determined experimentally, will also fall below. The limit value for the resolution will be determined by the formula

$$p_{lim}^{tr} = \frac{p^{tr}}{N_{lim}} = 1. \tag{4}$$

If this fraction is less than 1, then the threshold value is less than the resolution limit value, and such a threshold cannot be detected.

3. Experiment to test the hypothesis

Theoretical studies on the determination of thresholds are to be further verified using experimental tests. When testing, a secure local network, connected to the Internet, have to be attacked from the external IP addresses using the most common attack tools. During an attack, the values of the variables, used to detect the attacks, are to be recorded so that they can be compared with theoretically found thresholds.

Thus, there is the relation to be experimentally verified:

$$p_k^{tr} N_k = \text{const}, \tag{5}$$

where p_k^{tr} is the network value measured during the attack at the sampling frequency of N_k, which will change during the experiment. At the same time, the intensity of the attack, set during the experiment using the LOIC (Low Orbit Ion Cannon) utility, remains unchanged.

To conduct the experiment, a special experimental network complex, consisting of an Aruba 2930F switch, a honeypot as an attack target, and an sFlow agent, was created (Fig. 3). In this case, when the sampled traffic analysis technology (available on the Aruba switch) is enabled, only one of the N packets, passing through the switch, is transmitted to the device that aggregates the information (this is the sFlow agent, where the traffic is collected and analyzed).

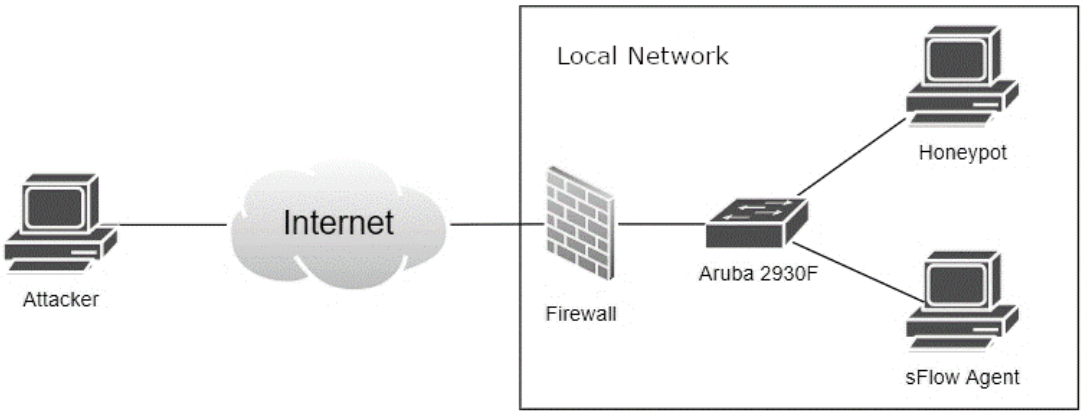

Fig. 3. Scheme of the network complex

This complex was installed in the global network, and all the devices received their public IP addresses. During the preparation of the network complex for the experiment, it was necessary to gain access to the local network in which the complex was located for control. To do this, the provider was requested to have a static public IP address added to the allowed list of incoming connections. Packet sampling on the switch was configured using the command

```
sflow 1 sampling 24 100
```

This command uses the following options:

- 1 — the sFlow agent port;
- 24 — the port to analyze traffic from;
- 100 — the sampling parameter (1 packet out of 100).

Traffic aggregation on the sFlow agent was carried out using the sflowtool utility with the command

```
sflowtool -p 6343 -J
```

In this command, 6343 is the port that receives traffic from the switch.

The training attack was carried out using the LOIC program, an open-source program designed to carry out DoS attacks[1]. This program is the de facto standard for conducting

[1] *Batishchev A. M.* LOIC (Low Orbit Ion Cannon). 2004. Available at: http://sourceforge.net/projects/loic (accessed February 18, 2023).

intrusion testing and allows the carrying out of most types of existing attacks; moreover, it is constantly improved, and new features are added.

To start the attack, configuration, which involves setting the target URL or IP address, attack speed, method, port, number of flows, and waiting timeout, is required. After setting all the necessary parameters, the attack can be launched with the "IMMA CHARGIN MAH LAZER" button (to stop the attack, the same button is used).

The attack was repeated several times for different samples, while all the attack parameters, entered in LOIC, were being saved. At the same time, the resolution for traffic on Aruba was changing. The samples "1 out of 50", "1 out of 100", "1 out of 200", "1 out of 500", and "1 out of 1000" were used; the traffic during the attack was recorded using the sFlow agent.

4. Analysis of the obtained results

During the attack, the sFlow agent collects traffic data in a JSON file. The information from the file requires additional processing, structuring, and generalization for the analysis. To do this, a script, which allows visual interpreting of the data using graphs, was written (the Python programming language was used).

Figure 4 shows an example of the traffic data recorded by the sFlow agent and processed by the script.

Traffic processing allows for determining the various characteristics of the data flow, coming from a single external IP address, at any time during the experiment. For example, it is possible to determine the number of received packets per second or the bit rate for incoming TCP/UDP traffic, as well as build their dependence on time. The time dependence diagram for the downstream packet rate $B_N(t)$ for the sample "1 out of 50" is shown in Fig. 5.

The abscissa shows the time elapsed since the beginning of the experiment, and the ordinate shows the rate of the downstream packet rate $B_N(t)$ measured in the number of flows per second.

For comparison, the time dependence diagram for the downstream packet rate $B_N(t)$ for the sample "1 out of 200" is shown in Fig. 6.

The traffic data collected allow finding average values and their variations for network parameters, as well as testing the hypothesis of a constant value for the ratio from (5). The data processed is presented in the Table.

Table. Incoming traffic data

Sl. No.	N_k	B_k (packets per second)	$\sigma(B_k)$	$B_k \cdot N_k$	$\sigma(B_k) \cdot N_k$
1	50	44.50	6.02	2225	301
2	100	15.90	2.79	1590	279
3	200	9.77	2.33	1954	466
4	500	4.19	0.74	2095	370
5	1000	3.65	0.70	3650	700

The fifth column of the Table shows data the product of the downstream packet rate $B_N(t)$ and the sample size N_k. Considering the measurement error from column 6, this ratio can be considered constant. For a more visual demonstration of the constant ratio, there is a graph in Fig. 7.

The abscissa is the sample size N_k, and the y-axis is the downstream packet rate B_N measured in flows per second. The graph in Fig. 7 clearly shows that there is a straight line within the two-fold mean-square error $\sigma(B_k)$ that is, the hypothesis (Section 3) was experimentally verified.

```
{
 "datagramSourceIP":"192.168.1.2",
 "datagramSize":"1256",
 "unixSecondsUTC":"1634213701",
 "localtime":"2021-10-14T16:15:01+0400",
 "datagramVersion":"5",
 "agentSubId":"0",
 "agent":"91.222.129.200",
 "packetSequenceNo":"3609022",
 "sysUpTime":"3061563254",
 "samplesInPacket":"6",
 "samples":[{
   "sampleType_tag":"0:1",
   "sampleType":"FLOWSAMPLE",
   "sampleSequenceNo":"9689728",
   "sourceId":"0:2",
   "meanSkipCount":"50",
   "samplePool":"513989805",
   "dropEvents":"393004",
   "inputPort":"1",
   "outputPort":"2",
   "elements":[{
     "flowBlock_tag":"0:1",
     "flowSampleType":"HEADER",
     "headerProtocol":"1",
     "sampledPacketSize":"536",
     "strippedBytes":"4",
     "headerLen":"128",
     "headerBytes":"00-21-5E-F0-2A-A8-52-54-00-BA-F0-87-
08-00-45-00-02-06-AD-95-00-00-40-11-53-22-5B-DE-80-C8-
4A-D0-50-B9-13-C4-1B-14-01-F2-52-41-53-49-50-2F-32-2E-30
-20-34-30-33-20-46-6F-72-62-69-64-64-65-6E-0D-0A-56-69-
61-3A-20-53-49-50-2F-32-2E-30-2F-55-44-50-20-37-34-2E-32
-30-38-2E-38-30-2E-31-38-35-3A-36-39-33-32-3B-62-72-61-
6E-63-68-3D-7A-39-68-47-34-62-4B-2D-32-30-38-35-32-31-38
-37-35-30-3B-72",
     "dstMAC":"00215ef02aa8",
     "srcMAC":"525400baf087",
     "IPSize":"518",
     "ip.tot_len":"518",
     "srcIP":"91.222.128.200",
     "dstIP":"74.208.80.185",
     "IPProtocol":"17",
```


Fig. 4. Sample data for the captured traffic

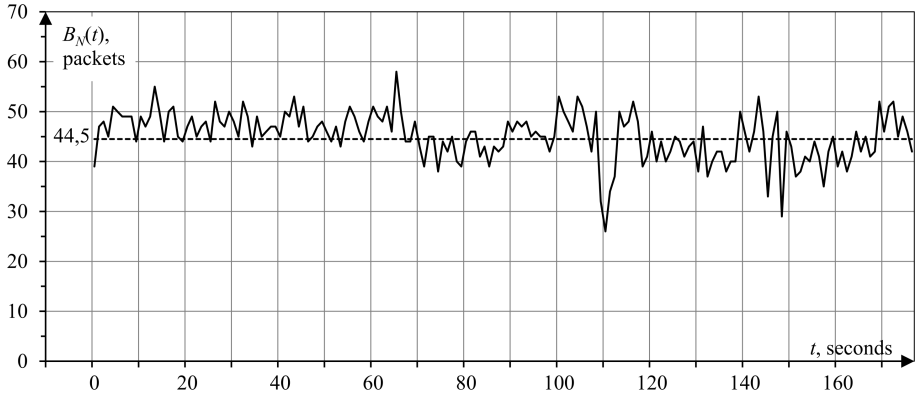

Fig. 5. Time dependence for the downstream packet rate $B_N(t)$ “1 out of 50”

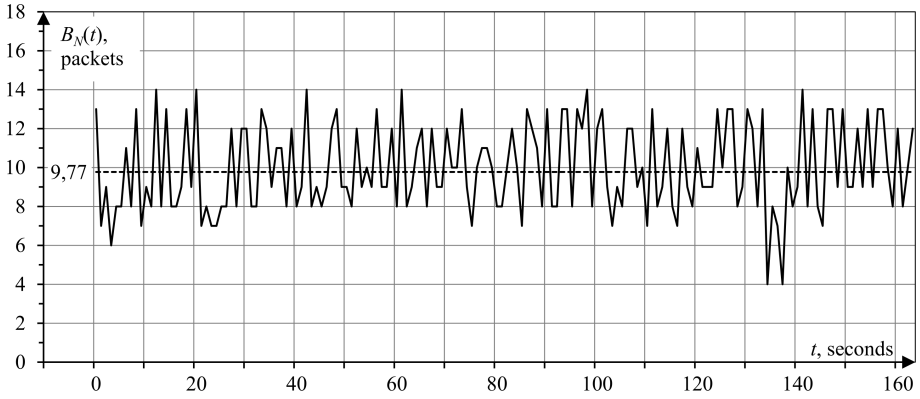

Fig. 6. Time dependence for the downstream packet rate $B_N(t)$ “1 out of 200”

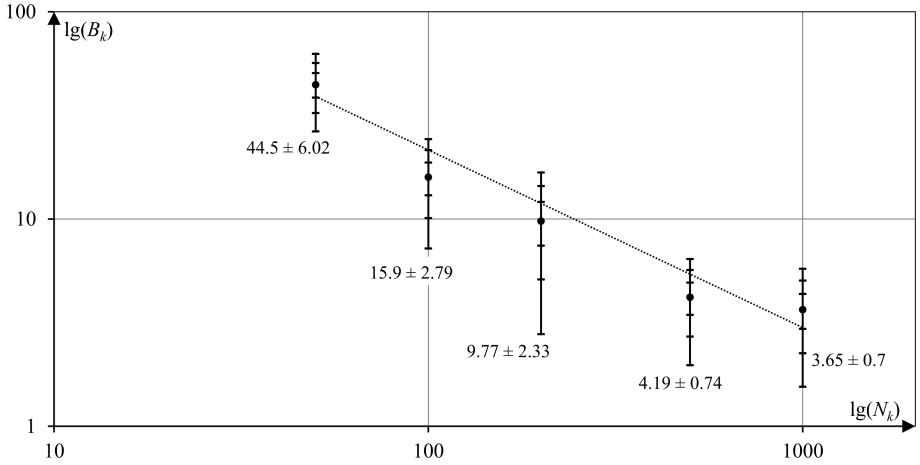

Fig. 7. Dependence of the downstream packet rate $B_N(t)$ on the sample size N_k in the logarithmic axes

Conclusions

This work defines the rules for finding threshold values for the main network variables used to detect network intrusions under conditions of limited data sampling. Such variables include the number of active flows and incoming TCP and UDP traffic generated by a single external IP address.

The peculiarity of this solution is that the traffic analysis is performed using the sFlow technology. This technology involves limited packet sampling, and its parameters can be changed (1 packet out of 50 can be analyzed, but this value can reach 1000).

When working with incomplete data, it is important to solve the problem of measuring the threshold values for variables, used for the network intrusion detection, and it was solved. Since a small number of traffic packets was initially analyzed, the measured values changed too. The main conclusion is that the product of threshold and sample resolution remains constant. Thus, it is possible to link not only the NetFlow and sFlow data but also threshold values obtained at different sample resolutions. In addition, this work defines the size of the maximum resolution, at which an attack with a given threshold can be detected.

The theoretical research, focusing on the determination of thresholds, was experimentally verified. Based on the obtained theoretical results, a security infrastructure, measuring the values of the variables used for intrusion detection and comparing them with threshold values, was developed.

While testing, a local network, connected to the Internet, was subjected to DDoS attacks from the external IP addresses using the most common attack tools. During the attack, the values of the variables, used to detect the attacks, were fixed at different sample resolutions from 50 to 1000 (using 100, 200, and 500 as intermediate values).

During the experiment, the traffic from the attack addresses was recorded, and its average value and standard deviation were found. Based on the data collected, the hypothesis of a constant value of the product of the observed threshold value and the sample size was tested. Considering the experimental error, this hypothesis was verified.

The novelty of this approach is that it is not the entire incoming traffic that is analyzed but the limited sample of incoming traffic packets. At the same time, the values of variables, that exceed the threshold, are also confidently fixed, and only small flows that do not affect the selection of attack IP addresses are discarded. The formulated hypothesis established the new threshold values, as well as the ways to measure them using the data from a limited sample of packets.

References

1. Sukhov A. M., Sagatov E. S., Baskakov A. V. Rank distribution for determining the threshold values of network variables and the analysis of DDoS attacks. *Procedia Engineering*, 2017, vol. 201, pp. 417–427. https://doi.org/10.1016/j.proeng.2017.09.666
2. Claise B. *Cisco systems netflow services export version 9*. 2004. https://doi.org/10.17487/rfc3954
3. Giotis K., Argyropoulos C., Androulidakis G., Kalogeras D., Maglaris V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Computer Networks*, 2014, vol. 62, pp. 122–136. https://doi.org/10.1016/j.bjp.2013.10.014
4. Li B., Springer J., Bebis G., Gunes M. H. A survey of network flow applications. *Journal of Network and Computer Applications*, 2013, vol. 36, iss. 2, pp. 567–581. https://doi.org/10.1016/j.jnca.2012.12.020
5. Feinstein L., Schnackenberg D., Balupari R., Kindred D. Statistical approaches to DDoS attack detection and response. In: *Proceedings DARPA Information Survivability Conference and Exposition*. Washington, DC, USA, 2003, vol. 1, pp. 303–314. https://doi.org/10.1109/DISCEX.2003.1194894
6. David J., Thomas C. DDoS attack detection using fast entropy approach on flow-based network traffic. *Procedia Computer Science*, 2015, vol. 50, pp. 30–36. https://doi.org/10.1016/j.procs.2015.04.007
7. David J., Thomas C. Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic. *Computers & Security*, 2019, vol. 82, pp. 284–295. https://doi.org/10.1016/j.cose.2019.01.002
8. Singh K., Dhindsa K. S., Nehra D. T-CAD: A threshold based collaborative DDoS attack detection in multiple autonomous systems. *Journal of Information Security and Applications*, 2020, vol. 51, art. 102457. https://doi.org/10.1016/j.jisa.2020.102457
9. Garcia-Teodoro P., Diaz-Verdejo J., Maciá-Fernández G., Vázquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 2009, vol. 28, iss. 1–2, pp. 18–28. https://doi.org/10.1016/j.cose.2008.08.003
10. Patel S. K., Sonker A. Rule-based network intrusion detection system for port scanning with efficient port scan detection rules using snort. *International Journal of Future Generation Communication and Networking*, 2016, vol. 9, iss. 6, pp. 339–350. https://doi.org/10.14257/ijfgcn.2016.9.6.32
11. D'Cruze H., Wang P., Sbeit R. O., Ray A. A software-defined networking (SDN) approach to mitigating DDoS attacks. In: Latifi S. (ed.) *Information Technology – New Generations*, Advances in Intelligent Systems and Computing, vol. 558. Springer, Cham, 2018, pp. 141–145. https://doi.org/10.1007/978-3-319-54978-1_19
12. Bekeneva Ya. A. Analysis of DDoS-attacks topical types and protection methods against them. *Proceedings of Saint Petersburg Electrotechnical University Journal*, 2016, vol. 1, pp. 7–14 (in Russian). EDN: TGYPJD
13. Zakharov A. A., Popov E. F., Fuchko M. M. SDN architecture, cyber security aspects. *Vestnik SibGUTI*, 2016, iss. 1, pp. 83–92 (in Russian). EDN: WLSRVP

14. Glassman S. A caching relay for the world wide web. *Computer Networks and ISDN Systems*, 1994, vol. 27, iss. 2, pp. 165–173. https://doi.org/10.1016/0169-7552(94)90130-9
15. Wang D., Cheng H., Wang P., Huang X., Jian G. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 2017, vol. 12, iss. 11, pp. 2776–2791. https://doi.org/10.1109/TIFS.2017.2721359
16. Zhang S., Sun W., Liu J., Nei K. Physical layer security in large-scale probabilistic caching: Analysis and optimization. *IEEE Communications Letters*, 2019, vol. 23, iss. 9, pp. 1484–1487. https://doi.org/10.1109/LCOMM.2019.2926967

Поступила в редакцию / Received 21.03.2023
Принята к публикации / Accepted 29.05.2023
Опубликована / Published 30.08.2024