Development and Investigation of Multi-Cloud Platform Network Security Algorithms Based on the Technology of Virtualization Network Functions[1]

I. Bolodurina
Department of Applied Mathematics
Orenburg State University
Orenburg, Russia
E-mail: prmat@mail.osu.ru

D. Parfenov
Faculty of Distance Learning Technologies
Orenburg State University
Orenburg, Russia
E-mail: fdot_it@mail.osu.ru

V. Torchin
Department of Applied Mathematics
Orenburg State University
Orenburg, Russia
E-mail: vadim.torchin@gmail.com

L. Legashev
Department of Applied Mathematics
Orenburg State University
Orenburg, Russia
E-mail: silentgir@gmail.com

Abstract — **The aim of the research is to increase the effectiveness of firewalling facilities by means of conflict-free optimization of security rules and application of the neural network approach in software-defined networks. In the framework of the study, based on the joint use of intelligent mathematical approaches and modern technology of virtual network functions, it was possible to increase the productivity and security of the resources of the enterprise cloud platform. The results obtained experimentally make it possible to show a conclusion about the effectiveness of the practical application of the approach developed in the work. The application of the developed approach gives an improvement in the criterion of the average response time for the request, as well as the load on the central processor of the firewall.**

Keywords — adaptive firewall, SDN, software-defined infrastructure, multi-cloud platforms, neural network, network function virtualization, cyber security

I. Introduction

There are currently actively developing the market for telecommunication services. The large enterprise companies for more effective business work are leasing virtual data centers to host their own IT infrastructure. Therefore, the most popular segment of such services is providing for users network services based on multi-cloud platforms. The popularity of using such technical solutions has led to the fact that users and providers of telecommunication services face daily challenges related to threats in the field of cybersecurity.

According to the analysis of the leading suppliers of network equipment, such as Cisco and Huawei, the number of active threats to cyber security increases annually by 15–25%. Assessing the vector of attacks on the IT infrastructure that supports the operation of information systems in enterprise companies, we can see the rating of the following threats: restriction of access of legitimate users to the key resources of the company (25%); disruption of work of technological equipment (35%); obtaining unauthorized access to official or confidential information, as well as its intentional or accidental disclosure, distortion, or destruction due to the violation of the company's security policy (45%). In practice, the list of cyber attacks aimed at the corporate network can be divided into four main groups. It includes attacks denial of service (DDoS) attack, Remote to Local (User) attack (R2L), User to root (U2R) attack and Probing attack [2].

To prevent active threats, providers need effective tools for monitoring processes in the network, monitor services hosted in the network, as well as proactive control of security elements. Today the most popular and effective approach to networking for the provision of services based on virtual data centers is the use of software-defined network technologies (SDN). The use of this technology is due to a number of advantages. First of all, SDN greatly simplifies the design and operation of the network, since

[1] The research work was funded by RFBR, according to the research projects No. 16–37–60086 mol_a_dk, 16–07–01004, 18–07–01446, 18–47–560016 and the President of the Russian Federation within the grant for state support of young Russian scientists (MK-1624.2017.9).

978-1-5386-9456-5/18/$31.00 ©2018 IEEE

it allows centralized intelligent control at the controller level. Secondly, SDN allows network administrators to quickly configure and optimize network resources based on an aggregated set of data collected in a single location. Thirdly, the use of SDN allows providing protection by dynamically analyzing data flows circulating in a virtual data center [14].

Another technology used to organize a network based on virtual data centers is the technology of network function virtualization (NFV). Technology NFV offers a new way of designing, deploying network services based on a multi-cloud platform. Virtualization of network functions allows to separate network functions, such as NAT, firewall, IDS, IPS, DNS from the hardware level [13]. In addition, it allows you to consolidate all the network components necessary to support virtualized infrastructure at the software level. In addition, it allows you to consolidate all the network components necessary to support virtualized infrastructure at the software level. Like SDN, NFV also offers advantages when designing a secure network environment in a virtual data center. One of the main advantages of NFV technology is scalability. To quickly meet the dynamically changing needs of users and provide new services, telecommunications service providers should be able to adapt their network architecture without changing the hardware.

Software-defined network and network function virtualization technologies have much in common with each other: they relate to the technologies of the next generation computer networks, can coexist in the same network environment, have many common characteristics and components. Therefore, in the framework of this study, we proposed a solution based on the hybrid use of SDN and NFV for organizing network security for a multi-cloud platform deployed on the basis of a virtual data center.

The aim of the research is to increase the effectiveness of firewalling facilities by means of conflict-free optimization of security rules and application of the neural network approach in software-defined networks.

The other part of the paper is organized as follows. Section II is devoted to describing traditional approaches to cyber security and solution proposed by world scientists for solving this problem. In section III we describe the methods and approaches applied within a framework of our solution. We provide describes the model of adaptive creation rules for firewalls and provides describes an optimization approaches based on the neural network and genetic algorithm. Section IV provides experimental results of our investigation. Conclusion section includes the summary of our investigation as well as future work overview.

II. Related work

Today there are many approaches to ensuring the security of applications and services, including the use of technologies for software-defined networks (SDN) and multi-cloud platforms.

In article [1], the problem of responding to an incident in the cloud is considered. An incident is an approach to eliminating and control the consequences of a security breach or a network attack. As a key criterion of the model authors select the minimization of the incident processing time is considered due to the introduction of security controllers and security domain in the cloud infrastructure for the analysis of network threats.

To counter network cyber attacks, technologies began to appear to provide continuous monitoring on any device connected to the network in order to increase fault tolerance by detecting and mitigating targeted threats. The authors of the research [2] describe the Gestalt security architecture, which is based on the principles of strong isolation, the policy of least privileges, the concept of providing information security (defense-in-depth), cryptographic authentication, encryption and self-healing. Remote monitoring is carried out through an organized workflow through a multitude of components connected by a specialized secure communication protocol that together provide secure and sustainable access.

Conventional firewalls are used to enforce network security policies on the boundaries within the network. In [3], the authors take advantage of the SDN to turn the network infrastructure into a virtual firewall, thereby improving network security. The virtual firewall as ACLSwitch is presented, which uses the OpenFlow protocol to filter network traffic between OpenFlow switches. The authors also introduce domain policies that allow the use of different filtering configurations for different network switches.

The optimal use of IDS and IPS systems is also presented in [4]. The authors investigated the open source snort system. The implementation, tuning, installation of the system and the problems arising during the research are studied.

Firewalls often have vulnerabilities that can be exploited by cybercriminals. The publication [5] is devoted to the investigation of some possible fingerprinting methods (fingerprint, identification) of the firewall, which turned out to be quite accurate. The authors also studied denial of firewalling attacks in which attackers use carefully processed traffic to overload the firewall. In [6] the implementation of a third-level firewall with a full-mesh topology with 1 controller and 6 switches is presented. Modification of the learning switch code for the POX controller is performed for the full-mesh topology inside the Mininet network emulator. In this case, the packet flow between hosts is controlled in accordance with the rules recorded in the learning switch through the OpenFlow controller.

In the context of the joint use of SDN and NFV, the NetFATE architecture is presented in [7], a platform

designed for placing virtual network functions on the network boundary. This architecture is based on free open-source software on the provider's nodes and client equipment, which leads to a simpler deployment of functions and lower management costs.

An important part of any network architecture is application identification, which contains information about the activity of network applications, and their use of bandwidth. The article [8] focuses on the promotion of open source technology to identify applications. This technology is presented as the future of firewalls, where the administrator can view more detailed information about network traffic compared to current approaches.

The issues of achieving elasticity for network firewalls are discussed in detail in [9]. Elasticity here means the ability to adapt to network load changes by releasing and allocating resources in an autonomous manner. The elasticity of cloud firewalls is aimed at satisfying a consistent performance evaluation using the minimum number of instances of the firewall. The author's contribution of the publication is to determine the number of instances that must be dynamically adjusted in accordance with the load of incoming traffic and the base of firewall rules.

Due to software processing of network functions, the performance of NFV significantly decreases depending on the types of NFV and the configuration of NFV applications. In the publication [10], the authors pay special attention to the analysis of the virtual firewall as a representative of NFV. The paper proposes a method for estimating the latent load of a virtual firewall using rules in the ACL and the amount of traffic for each rule.

A review of the related research has shown that to effectively build a list of rules for a firewall, two tasks must be solved:

- automation of rules for the firewall based on traffic data circulating in the network;
- to optimize the obtained list of rules for the firewall.

To solve the first task, it is necessary to perform classifications of network traffic passing through the corporate network. To solve the second problem, we use the iterative method in which we distinguish two main stages. At the first stage, we will perform the clustering of the rules obtained as a result of solving the first task, and the subsequent deducing of the rules from the clusters. In the second stage, to solve the second problem, we apply the algorithm of conflict-free optimization of the list of firewall rules.

III. The model of adaptive creation rules for the firewalls

To implement the presented plan, it is first of all necessary to determine the model of rules for the firewall used to secure the corporate multi-cloud platform.

The firewall rule is usually a string consisting of certain characteristics of a network connection and a decision about the admissibility of such a connection. Within the framework of the study, the following characteristics were chosen from a variety of characteristics describing network connections: the source and destination IP address, the corresponding ports and the protocol through which the data is transmitted. When you select these characteristics, the firewall rule in general form will look like:

$$<\text{rn, id\_src\_ip, id\_dst\_ip, src\_p, dst\_p, p\_id, tg, p\_cnt}>, \quad (1)$$

where rn — rule number in the list; id_src_ip — ID of the IP address of the packet sender from the IP-address of the network controller; dst_ip — ID of the IP address of the packet receiver from the IP-address of the network controller; sr_p — port of the sender of the packet; dst_p — port of the receiver of the packet; p_id — network protocol through which the connection is made; tg — the resolution on access or denial of the connection; p_cnt — number of packets of traffic.

A list of records about packages is represented as a set of the following form:

$$X = \{x_k\},\ k = \overline{1, n}, \quad (2)$$

where n — the length of the list of firewall rules for a corporate multi-cloud platform.

Define the range of possible values for the elements of a given vector, namely the characteristics of the traffic for which the filtered data will be filtered in the network of the software-defined infrastructure. They are represented as a list of lines of the form (1). Each such string is represented as a vector:

$$x_k = \{x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}\}, \quad (3)$$

where k — number in the list; $x_{k1} \in [0;\ 2^{32} - 1]$ — IP-address of the sender of the packet; $x_{k2} \in [0;\ 2^{32} - 1]$ — IP address of the receiver of the packet; $x_{k3} \in [0;\ 65535]$ — port of the sender of the packet; $x_{k4} \in [0;\ 65535]$ — port of the receiver of the packet; $x_{k5} \in \{0,\ 1,\ 2\}$ — permissible values of the identifier of the network protocol through which the connection is made.

The next step, let's present a list of firewall rules, where the rules correspond to the model (2) as a set:

$$R = \{r_i: r_i = \{r_{i,1}, r_{i,2}, r_{i,3}, r_{i,4}, r_{i,5}, r_{i,6}, r_{i,7}, r_{i,8}\}\}, \quad (4)$$

where $r_{i1} \in [0;\ m]$ — number in the list; $r_{i2} \in [0;\ 2^{32} - 1]$ — IP-address of the sender of the packet; $r_{i3} \in [0;\ 2^{32} - 1]$ — IP address of the receiver of the packet; $r_{i4} \in [0;\ 65535]$ — port of the sender of the packet; $r_{i5} \in [0;\ 65535]$ — network protocol through which the connection is made; $r_{i6} \in N$ — number of the protocol symbol stored in the DB of the network controller; $r_{i7} \in \{0;\ 1\}$ — the decision on admissibility or inadmissibility of connection, where 0 — connection is prohibited, 1 — connection is allowed; $r_{i8} \in N$ — number of packets of traffic.

After reducing the input data to the required form, we formulate the statement of the problem. Given a lot of records about the traffic passing through the corporate network in the form of a set of the type (3), it is required to build a set of non-conflicting rules as type (4):

$$R = \{r_i\},\ i = \overline{1, m},\ m \to \min. \quad (5)$$

This work assumes the solution of the problem by the iterative method in two stages. The first step is the construction of the initial list of rules R_1 by classifying the set X into two classes. The second step consists of making of the set R_{opt} by clustering the set R_1 with the subsequent deduction of the rules from the clusters, that is, the construction of the set R_2 with the subsequent optimization of this set by the algorithm of conflict-free optimization, that is, the construction of the set R_{opt}.

Suppose given a set X of the form (4), it is required to construct a list of rules or a set R_1, that is, to compose a classifying function $f(x)$: $X \to R$.

There are many different methods for solving this problem. In the framework of this study, we will use the classification based on the neural network. This is due to the fact that classification is a classic task for neural network methods. The most applicable for solving such a problem is a multi-layered perceptron type architecture. The number of neurons in the input layer is calculated depending on the input data. In this case, the size of the input layer of the neural network will be 99 neurons. It takes 32 bits to write an IP address, and 16 bits to write ports, the number of protocols considered is 7, therefore, 3 bits are required to write them. To train the selected neural network model, we will use the algorithm for back-propagation of the error.

A. Clustering firewall rules

One of the common types of attacks on computer networks is a constantly repeated attempt to gain access to a particular resource with the expectation that a typical packet of traffic of this attack on the network will satisfy a certain rule of the firewall. The purpose of the attack is to find the rule at the bottom of the list. As a result of such an attack, there is a rapid increase in the load on the processor and an increase in the amount of consumed RAM on the firewall. This leads to a drop in performance of the security system as a whole. In order to reduce memory costs and time to bypass the list of rules by a firewall, it is necessary to solve the problem of reducing the list of rules, without losing the characteristics responsible for protecting the corporate multi-cloud platform. For these purposes, it makes sense to break the rules into clusters in order to derive new, generalized rules from them.

Let's formulate the mathematical formulation of the rules of firewall clustering tasks.

Let there be given a set of rules $R = \{r_i\},\ i = \overline{1, m}$. It is required to compose a sample partition into disjoint subsets called clusters in such a way that each subset consists of objects that are close in some metric. It is necessary to compose a clustering function $f(r)$: $R \to Y$, that assigns to each element of the set an element of the set $Y = \{y_1, y_2, ...\}$ — the set of cluster numbers.

An important aspect in solving the clustering problem is the choice of the distance function, or metric. The metric is a measure of proximity, which is algorithms. As part of the study, the Euclidean distance was taken as the metric by the following characteristics: the source and destination IP addresses, the corresponding ports, the protocol by which the connection is made and the decision on the admissibility of the connection. Thus, the formula for the distance function has the following form:

$$D(r_1, r_2) = \sqrt{a(r_{1,2} - r_{2,2})^2 + b(r_{1,3} - r_{2,3})^2 + c(r_{1,5} - r_{2,5})^2 + d(r_{1,7} - r_{2,7})^2}. \quad (6)$$

This function was used with empirically selected parameters $a = 0.55$; $b = 0.55$, $c = 38745.6$; $d = 2^4 - 1$.

B. Algorithm for deducing firewall rules from clusters

The next step is to remove the rules from the clustered list of rules. Input data will be a list of rules, broken into clusters R_{kl}. The output of the algorithm is expected to list the generalized rules R_{opt}. In the framework of this study, an algorithm was developed. The list of clusters is denoted by u. Then for all clusters from u, we have:

Input: Clustered rule list R_{kl}, list of clusters u

Output: Generalized list of firewall rules R_{opt}

$$r_{i,1} = \frac{\min(r_{k,1})}{\text{mask}} = 32 - \log_2(\max(r_{k,1}) - \min(r_{k,1}));$$

$$r_{i,2} = \frac{\min(r_{k,2})}{\text{mask}} = 32 - \log_2(\max(r_{k,2}) - \min(r_{k,2}));$$

$$r_{i,3} = \{r_{1,3}, ..., r_{k,3}\};$$

$$r_{i,4} = \{r_{1,4}, ..., r_{k,4}\};$$

$$r_{i,5} = \{r_{1,5}, ..., r_{k,5}\};$$

$$r_{i,6} = r_{i,6}.$$

C. Algorithm for optimizing the list of rules by ranking sorting

An important parameter for the list of rules, in addition to the breadth of coverage of the protected resources and the value of the list, is also the arrangement of rules. It is possible to protect the corporate network from this kind of attacks by sorting the list of rules in order to place the most frequently used rules in the top of the list in order to save the system from having to go through more rules, thus saving both machine time and load on the device that performs the functions of the firewall.

To solve this problem, the following algorithm was developed. Let's describe it.

Input:

- A set $\boldsymbol{X}$ of traffic headers, passing by the rule "Deny any, any";
- Non-conflicting list of rules $\boldsymbol{R} = \{\boldsymbol{R}_i\}$, $i = 1, n$, where $\boldsymbol{Rn}$ — "Deny any, any";

Step 1: To assign each rule and each element of the set $\boldsymbol{X}$, the weight by formula $w_i = \frac{k_i}{k}$, where k_i — the number of traffic passes by the rule i, k — total number of packets passing through the network.

Step 2: **IF** $w_n > w^*$ **THEN go to Step 3**, **ELSE go to Step 5**.

Step 3: Create a Deny Rule $\boldsymbol{R}_{n-1}$, according to the element $\boldsymbol{X}$.

Step 4: Sort the set $\boldsymbol{R}$ by its value, put the rule "Deny any, any" in the last place. **Go to Step 2.**

Step 5: End.

This algorithm does not reduce the size of the list but allows you to make a more optimal list of rules. At the same time, there is protection against attacks aimed at the occurrence of failures in the work of firewall.

D. Genetic algorithm for deducing firewall rules

To compare the effectiveness of the proposed solution within the framework of this study, we also developed an approach that allows us to optimize the list of network security rules based on the genetic algorithm. This work assumes the solution of the problem by an iterative method in two stages:

1) *Generation of the set of the most optimal rules by the sorting criterion for the parameter p_{cnt} on the basis of the genetic algorithm*.

2) *Dropping "bad" rules in accordance with the specified QoS policies*.

As an individual, one of the rules (4) of firewalling is adopted. The size of the initial population corresponds to the number of rules m in the list. We represent the given rules in the form of columns of the matrix M_{init}:

$$M_{\text{init}} = \begin{pmatrix} r_{1,2} & \cdots & r_{m,2} \\ r_{1,3} & \cdots & r_{m,3} \\ r_{1,4} & \cdots & r_{m,4} \\ r_{1,5} & \cdots & r_{m,5} \\ r_{1,6} & \cdots & r_{m,6} \\ r_{1,7} & \cdots & r_{m,7} \\ r_{1,8} & \cdots & r_{m,8} \end{pmatrix}. \quad (7)$$

Let us consider in more detail the operations of the genetic algorithm.

1) *The crossing operation consists in the random selection of two firewall rules from the M_{init} matrix and the formation of a new rule as follows*:

a) *The IP address identifier and port number of the sender are selected randomly*.

b) *The IP address identifier and port number of the recipient are equally likely to be inherited from one of the parent rules*.

c) *Protocol identifier number randomly takes one of three values* {0, 1, 2}.

d) *The decision on the admissibility or inadmissibility of the generated connection is specified in accordance with the table of rules* (3).

e) *The number of packets of traffic passing is calculated as follows*: *summation of packages of parental rules in case of their admissibility*.

Each new rule is added to the M_{init} matrix. The cross ratio is 100%. This means that with the initial number of rules m, the size of the final matrix after the cross operation will be $2m$.

2) *The mutation operation consists in randomly selecting one rule from the extended M_{init} matrix and changing the set of parameters of this rule by analogy with the crossing operation*.

3) *The selection operation consists in the sorting by non-increasing of the columns of the matrix on the basis of the criterion of the number of packets of the transmitted traffic*. *As a result of the selection operation*, *the m best individuals will be used as inputs to subsequent iterations of the work of the genetic algorithm*.

4) *Increase in the counter of epochs t++*.

Steps 1–4. Repeat until the specified number of epochs $t = E$ of the algorithm is reached.

Once we have generated a set of the most optimal firewall rules, the process of dropping the "bad" rules is performed in accordance with the specified QoS policies (5–10% of the rules are experimental). Excluding such rules from the final record table allows you to reduce the value of m, which in turn will reduce the load on the firewall CPU.

IV. Experimental results

Based on the proposed solution, the module implemented an adaptive firewall. The software is implemented as a virtual network function based on the Open Platform for NFV (OPNFV), assembled as a Docker container. For comparison, the traditional firewall, which is a packet filter, implemented inside the POX platform, was chosen.

Comparative analysis was carried out by comparing the results of the traditional firewall and the developed software module with the neural network. Before the experiment, the developed software module was trained. Also, we have developed a comparable set of rules for the traditional firewall, which allows for a correct comparison of the compared means. Within the framework of the study, work was evaluated under different loads for two

key indicators: response time in the network; load the central process on the firewall.

For carrying out load test of scenarios experimental attacks we create the virtual network in OpenStack cloud. It includes 4 OpenFlow switches (2 HP 3500yl, 2 Netgear GSM7200), 8 computing nodes (32Gb RAM, 4 cores), 1 server (32Gb RAM, 8 cores) with OpenFlow controller and 1 server (32Gb RAM, 4 cores) for monitoring function. As a selected fat tree topology with three levels. Routers connected compounds having the speed 1000 Mbit/s, and the computers are connected to a third level router via the second level network connections at 1000 Mbit/s. In this infrastructure was prepared 100 virtual machines (users nodes). Also, we include one node that controls the flows. We select random five attacked virtual machines (service host). The load is created using a special tool — hping3. It allows you to generate packets of a certain type in certain directions, which makes possible on different values of the number of packets, and as a consequence. The experiment consists of the sequential measurement of the indicators with a consecutive increase in the intensity of traffic flows in single-load frames in the range from 20 to 350Mbit/s

Response time is one of the key parameters when analyzing network performance and network resources. It shows how much time passes from the moment the request is sent by the user until the service responds to the request. In accordance with the experiment rules, the response time from the network was removed.

The list of security rules generated by the built-in firewall module under load, including avalanche, gives a response time difference of up to 20%, which indicates the effectiveness of the approach developed in the work. The load on the central processor of the shielding device is an important parameter when considering safety measures. This is an indication of whether the device is loaded, or inefficiently used.

The results show that an optimized set of rules built using the approach proposed in the study reduces the load on the firewall CPU.

V. Conclusion

The survey reviewed existing firewall solutions. A model for describing firewall security rules has been developed. Based on the constructed model, a new ap-

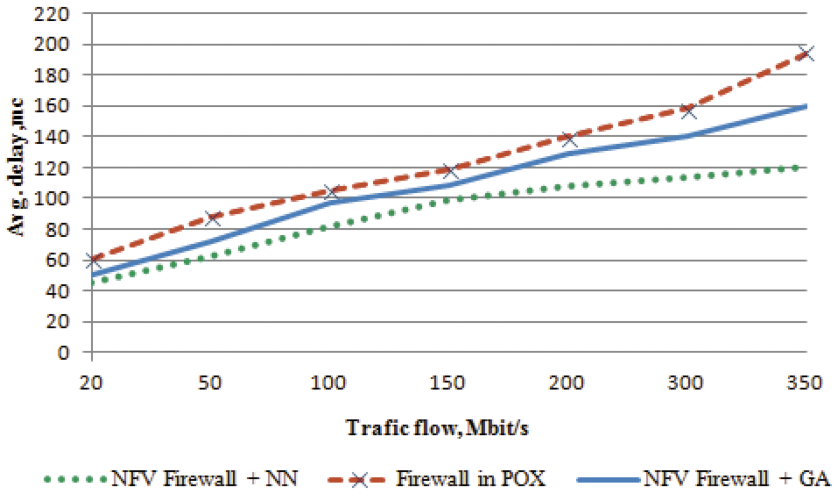

Fig. 1. The delay in the network for different speed of flows

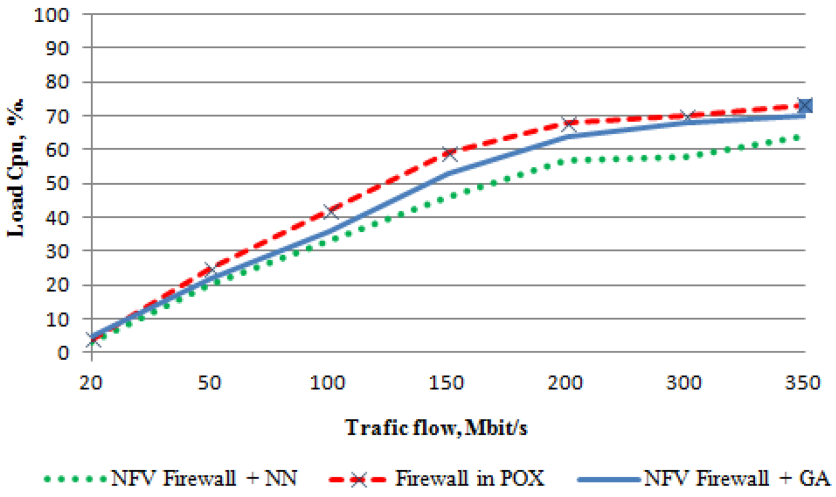

Fig. 2. The load CPU in the firewall for different speed of flows

proach was proposed and investigated to optimize the formation of the list of network security rules, based on the use of neural networks and the genetic algorithm. Two algorithmic solutions have been developed that realize the function of forming and optimizing rules on the firewall. To pre-select the list of rules, the neural network architecture was selected. As architecture, a hybrid artificial neural network consisting of two networks, namely a classifier based on a multi-layer perceptron and a cluster based on the Kohonen network, was chosen. During the development of algorithmic solutions, a comparative analysis of the classification and clustering algorithms was carried out. As a result, it is recommended to use the "multilayer perceptron" neural network classifier for automatic rule building, since it gives the best results in terms of performance. In the conducted research the campaign allowing to reduce the dimensionality of the list of rules of safety of a firewall by means of a network of Kohonen is offered. In the projected architecture, an algorithm for conflict-free optimization was implemented, which produces the final optimization by ranking and deducing the most frequently encountered exceptions from large prohibitive rules. This allows improving protection against attacks aimed at identifying security rules at the end of the list of firewalls and massive traffic generation according to these rules with the purpose of degrading the security system and increasing the number of failures. To compare the effectiveness of the proposed solution, a genetic algorithm was also developed and implemented that makes it possible to select and form an optimal list of network security rules. The algorithms and methods considered were implemented as a virtual network function for a software-defined network.

During the experimental research, it was revealed that the approach proposed by this study gives an increase in two key parameters, namely, response time, on average by 20%, with an increase in load and load of the central process of the shielding device, by an average of 4.5% load growth. Thus, the developed approach is effective for solving practical problems.

References

1. A. Adamov, and A. Carlsson, "Cloud incident response model," IEEE East-West Design Test Symposium. Yerevan, Armenia, pp. 1–3, 2016 [Proceedings of 2016 IEEE East-West Design Test Symposium (EWDTS), Yerevan, Armenia, 2016].
2. M. Atighetchi, and A. Adler, "A Framework for Resilient Remote Monitoring," International Symposium on Resilient Control Systems. Denver, USA, pp. 1–8, 2014 [Proceedings of 2014 7th International Symposium on Resilient Control Systems (ISRCS), Denver, USA, 2014].
3. J.N. Bakker, I. Welch, and K.G. Winston, "Seah Network-wide Virtual Firewall using SDN/OpenFlow," IEEE Conference on Network Function Virtualization and Software Defined Networks. Palo Alto, California, USA, pp. 1–7, 2016 [Proceedings of 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, California, USA, 2016].
4. M. Kashif, and Zahoor-ul-haq, "An Optimal Use of Intrusion Detection and Prevention System," European Intelligence and Security Informatics Conference. Manchester, United Kingdom, 2015 [Proceedings of 2015 European Intelligence and Security Informatics Conference (IDPS), Manchester, United Kingdom, 2015, P. 190].
5. A.X. Liu, A.R. Khakpour, J.W. Hulst, Z. Ge, D. Pei, and J. Wang, "Firewall Fingerprinting and Denial of Firewalling Attacks," in IEEE Transactions on Information Forensics and Security Journal, vol. 12, No. 7, pp. 1699–1712, 2017.
6. A. Kumar, and N.K. Srinath, "Implementing a firewall functionality for mesh networks using SDN controller," International Conference on Computational Systems and Information Systems for Sustainable Solutions. pp. 168–173, 2016 [Proceedings of 2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions, 2016].
7. A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene, "An Open Framework to Enable NetFATE (Network Functions At The Edge)," IEEE Conference on Network Softwarization. London, United Kingdom, pp. 1–6, 2015 [Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, United Kingdom 2015].
8. N.V. Patel, Dr. N.M. Patel, and C. Kleopa, "OpenAppID — Application Identification Framework," Online International Conference on Green Engineering and Technologies. Coimbatore, India, pp. 1–5, 2016 [Proceedings of 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 2016].
9. K. Salah, P. Calyam, and R. Boutaba, "Analytical Model for Elastic Scaling of Cloud-based Firewalls," in IEEE Transactions on Network and Service Management Journal, vol. 14, No. 1, pp. 136–146, 2016.
10. D. Suzuki, S. Imai, and T. Katagiri, "A New Index of Hidden Workload for Firewall Rule Processing on Virtual Machine," International Conference on Computing, Networking and Communications: Communications QoS and System Modeling. Santa Clara, California, USA, pp. 632–637, 2017 [Proceedings of 2017 International Conference on Computing, Networking and Communications (ICNC): Communications QoS and System Modeling, Santa Clara, California, USA, 2017].
11. D. Parfenov, and I. Bolodurina, "Methods and algorithms optimization of adaptive traffic control in the virtual data center," International Siberian Conference on Control and Communications. Astana, Kazakhstan, pp. 1–6, 2017 [2017 International Siberian Conference on Control and Communications (SIBCON2017), Proceedings 29–30 June 2017, Astana, Kazakhstan, 2017].
12. I.P. Bolodurina, and D.I. Parfenov, "Development and Research of Modelsof Organization Distributed Cloud Computing Based on the Softwaredefined Infrastructure," in Procedia Computer Science, vol. 103, pp. 569–576.
13. L. Qu, C. Assi, K. Shaban, and M. Khabbaz, "Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks," International Conference on Network and Service Management. Montreal, QC, Canada, pp. 153–159, 2016 [2016 12th International Conference on Network and Service Management (CNSM), 31 Oct.-4 Nov. 2016, Montreal, QC, Canada, 2016].
14. R.L. Smeliansky, "SDN for network security," International Science and Technology Conference. Moscow, Russia, pp. 1–5, 2014 [2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 28–29 Oct. 2014, Moscow, Russia, 2014].