Representation Model of Requests to Web Resources, Based on a Vector Space Model and Attributes of Requests for HTTP Protocol

Manh Thang Nguyen, Alexander Kozachok

***Abstract*— Recently, the number of incidents related to Web applications, due to the increase in the number of users of mobile devices, the development of the Internet of things, the expansion of many services and, as a consequence, the expansion of possible computer attacks. Malicious programs can be used to collect information about users, personal data and gaining access to Web resources or blocking them. The purpose of the study is to enhance the detection accuracy of computer attacks on Web applications. In the work, a model for presenting requests to Web resources, based on a vector space model and attributes of requests via the HTTP protocol is proposed. Previously carried out research allowed us to obtain an estimate of the detection accuracy as well as 96% for Web applications for the dataset KDD 99, vector-based query representation and a classifier based on model decision trees**

Tóm tắt – Trong những năm gần đây, số lượng sự cố liên quan đến các ứng dụng Web có xu hướng tăng lên do sự gia tăng số lượng người dùng thiết bị di động, sự phát triển của Internet cũng như sự mở rộng của nhiều dịch vụ của nó. Do đó càng làm tăng khả năng bị tấn công vào thiết bị di động của người dùng cũng như hệ thống máy tính. Mã độc thường được sử dụng để thu thập thông tin về người dùng, dữ liệu cá nhân nhạy cảm, truy cập vào tài nguyên Web hoặc phá hoại các tài nguyên này. Mục đích của nghiên cứu nhằm tăng cường độ chính xác phát hiện các cuộc tấn công máy tính vào các ứng dụng Web. Bài báo trình bày một mô hình biểu diễn các yêu cầu Web, dựa trên mô hình không gian vectơ và các thuộc tính của các yêu cầu đó sử dụng giao thức HTTP. So sánh với các nghiên cứu được thực hiện trước đây cho phép chúng tôi ước tính độ chính xác phát hiện xấp xỉ 96% cho các ứng dụng Web khi sử dụng bộ dữ liệu KDD 99 trong đào tạo cũng như phát hiện tấn công đi kèm với việc biểu diễn truy vấn dựa trên không gian vectơ và phân loại dựa trên mô hình cây quyết định.

***Keywords*— Computer attacks; Web resources, classification; machine learning; attributes; HTTP protocol.**

***Từ khóa*— Tấn công mạng; tài nguyên web, học máy, thuộc tính, giao thức HTTP.**

I. INTRODUCTION

Recently, the number of information security incidents has increased worldwide, related to the security of Web applications, due to the increase in the number of users of mobile devices, the development of the Internet of things, the expansion of many services and, as a result, the expansion of possible computer attacks.

The web resources of state structures and departments are also subject to attacks. One of the reasons for the growth of these attacks is also an increase in the number of malicious programs. Malicious programs can be used to collect information about users, personal data

and gaining access to Web resources or blocking them.

Impact on the rate of spread of various malware and viruses is caused by such factors as:

- widespread social networking;
- increased resilience and stealth botnets;
- cloud service distribution.

According to the analyses [1], attacks on Web applications account for more than half of all Internet traffic for information security. The purpose of the study is to improve the accuracy of detecting computer attacks on Web applications. The main result is the presented model for submitting requests to Web resources, based on the vector space model and attributes of requests via the HTTP protocol.

This manuscript is received June 14, 2019. It is commented on June 17, 2019 and is accepted on June 24, 2019 by the first reviewer. It is commented on June 16, 2019 and is accepted on June 25, 2019 by the second reviewer.

II. WAYS TO DETECT COMPUTER ATTACKS ON WEB APPLICATIONS

Many attack detection systems use 3 basic approaches: methods based on signature [2;3], anomaly detection methods [4–8] and machine learning methods.

A. Signature methods

The signature analysis based on the assumption that the attack scenario is known and an attempt to implement it can be detected in the event logs or by analyzing for network traffic with high reliability. There is a certain signature of attacks in the database of signatures.

Intrusion detection systems (IDS) that use signature analysis methods are designed to solve the indicated problem, as in most cases they allow not only detecting but also preventing the implementation of known attacks at the initial stage of its implementation. The disadvantage of this approach is the impossibility of detecting unknown attacks, the signatures of which are missing in the database of signatures.

B. Anomaly Detection Methods

Anomaly detection method is a way to detect a typical behavior of subjects in the world. At the same time in the system of detection of computer attacks models of ¬ the behavior of the subjects (behavior profiles) should be determined. For this purpose, test or training data sets are used to simulate traffic, which is considered legitimate in the network. For the operation of an attack detection system based on the detection of anomalies, it is necessary to develop a criterion for distinguishing the normal behavior of subjects from the anomalous. If the behavior deviates from normal one by an amount greater than a certain threshold value, then the system notifies of this deviation. Training datasets are also used to simulate malicious traffic so that the system can recognize patterns of unknown threats and attacks.

An important feature of the tasks of detecting atypical system behavior and detecting anomalies is the lack of a formal definition of the anomaly. It was obtained during the study, depending on the chosen method and the feature space.

For complex systems, while solving the problem of detecting anomalies, we should also apply machine learning methods and other data mining methods.

C. Anomaly detection methods using machine learning methods

Machine learning [9], as a section of artificial intelligence, is used as the emergence of anomalies, and the detection of abuse. This is explained by the fact that these approaches often use patterns of both normal and anomalous behavior of subjects as initial data for training.

1. Bayesian Networks

One of the most commonly used approaches to detect computer attacks is the Bayesian network. The Bayesian network [10] is a model that encodes the probabilistic relations between the events (variables) under consideration and provides some mechanism for calculating the conditional probabilities of their occurrence. A special case of this model is the naive Bayes classifier (Bayesian method) with strict assumptions concerning the independence of the input variables. Bayesian network [11; 12] - graph probabilistic model, which is a set of variables and their probabilistic dependencies according to Bayes.

In [13], pseudo-Bayesian evaluation functions are used to determine a priori and a posterior probability of new attacks. The authors argue that due to the properties of the proposed method, the system does not need prior knowledge of the patterns of new attacks. The authors used the "ADAM" system which consists of three modules:

- preprocessing module: to collect data from traffic and extract information on every connection;
- intellectual module: applies the rules of the association X→Y to the records of the connections, where X and Y, respectively, are the precondition and postcondition of the rules described inside the core of the system;

- classification module: new rules of association to normal or anomalous coexistence.

2. Neural networks

An artificial neural network is a mathematical model, as well as its software or hardware implementation, built on the principle of organization and functioning of biological neural networks – networks of cells of a living organism. From the point of view of machine learning, an artificial neural network is a special case of pattern recognition methods, discriminant analysis and clustering.

In [14], a neural network approach is described that combines the speed of processing network traffic by compressing features and the high accuracy of classifying network attacks. Detection of network attacks is associated with the release of a large number of signs by which classification can be made.

Evaluation of the effectiveness was carried out by the authors on the publicly available KDD99 base [15], containing about 5 million attack instances classified in 22 classes. To reduce the dimensionality of the attribute space, the authors use the method of main components and a recurrent neural network.

3. K-nearest neighbors

The k-nearest neighbor method (k-NN) [16] is a classification method, the basic principle of which is to assign an object of the class that is most common among the neighbors of this object. Neighbors are formed from a variety of objects which classes are already known. Based on the set value to k > 1, it is determined which of the classes to include the object being analyzed. If k = 1, then the object belongs to the class of the only nearest neighbor.

In [17], the authors used a combined approach – a combination of the genetic algorithm [18] and the k-nearest neighbor classifier to detect denial of service attacks. The goal of the genetic algorithm is to find the optimal weight vector, in which ω_i represents the weight of features $1 \le i \le n$. For two vectors features $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$ distance between them will be calculated as follows:

$$d(X,Y) = \sqrt{\omega_1(x_1 - y_1)^2 + \omega_2(x_2 - y_2)^2 + ... + \omega_n(x_n - y_n)^2} \quad (1)$$

After the evolution of the genetic algorithm at the training state, an optimal weight vector can be obtained, which leads to a better k-NN classification result. In the experiment, there were two datasets with 35 features: for learning (600 normal cases and 600 attacks) and testing (100 normal cases and 100 attacks). Detection accuracy of this developed approach was about 94.75%.

4. Method decision tree

Decision trees (also called classification trees or regression trees) are a decision support tool used in statistics and data analysis for predictive models.

Decision trees are tree structure of "leaves" and "branches". At the branches of the decision tree attributes are represented, in the "leaves" the values of the function are written, and in the remaining nodes the attributes are given by which the objects are distinguished. For classifying a new object, go down the tree from the root to the leaf and get corresponding class label according to classification rules based on values of attribute object.

The results of a comparative analysis of algorithms based on decision trees in relation to other algorithms are given in [19].

In [20], the authors proposed replacing the standard attack detection module in the system Snort with decision trees.

Experiments were performed on the DARPA data set and showed an increase in processing speed of pcap files used to analyze network packages, an average of 40.3% in comparison with the standard module.

In [21], a comparative analysis of the capabilities of an artificial neural network and the decision trees method for solving problems of detecting computer attacks is carried out. The researchers came to the conclusions that artificial neural network is effective for generalization and not suitable for detecting new attacks, while decision trees are effective for both tasks.

5. Support vector machine

The initial data in the support vector machine method is a set of elements located in space. The dimension of space corresponds to the number of classifying signs, their value determining the position of elements (points) in space.

The support vector machine method refers to linear classification methods. Two sets of points belonging to two different classes are separated by a hyperplane in space. At the same time, the hyperplane is constructed in such a way that the distances from it to the nearest instances of both classes (support vectors) were maximum, which ensures the strict accuracy of classification.

The support vector machine method allows [22; 23]:

- obtaining a classification function with a minimum upper estimate of the expected risk (level of classification error);
- using a linear classifier to work with nonlinearly shared data.

III. MODEL FOR PRESENTING REQUESTS TO WEB RESOURCES, BASED ON THE VECTOR SPACE MODEL AND ATTRIBUTES OF REQUESTS VIA HTTP

The anomaly detection approach is based on the analysis of HTTP requests processed by most common Web servers (for example, Apache or nginx) and is intended to be built in Web Application Firewall (WAF). WAF analyzes all requests coming to the Web server and makes decisions about their execution on the server (Fig.1).

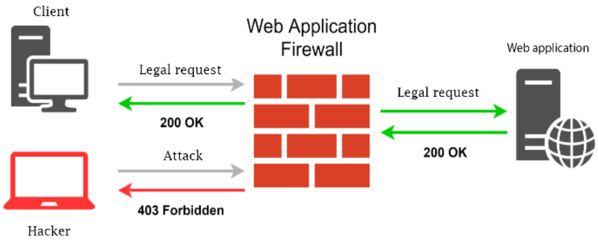


Fig.1. WAF in Web Application Security System

A. *Formation of feature space for our model*

To set the model for presenting requests to Web resources, the author has carried out the formation of a corresponding feature space, that has allowed to evaluate its adequacy from the standpoint of solving the problem of detecting computer attacks on Web applications.

In fig.2 the main stages of analyzing an HTTP request received at the Web server input are demonstrated. We divided the dataset into two parts: requests with information about attacks and normal requests. In the learning process, we will calculate all the necessary values such as the expected value and the variance of normal queries, then these values are stored in the database MySQL for the attack detection process. The analysis is performed on the appropriate fields of the protocol to ensure further possibility of its representation in the vector space model. It also analyzes and calculates a number of attributes selected by the author. Thus, the proposed query representation model allows moving from the text representation to the totality of features of the vector space model for the corresponding protocol fields and query attributes.

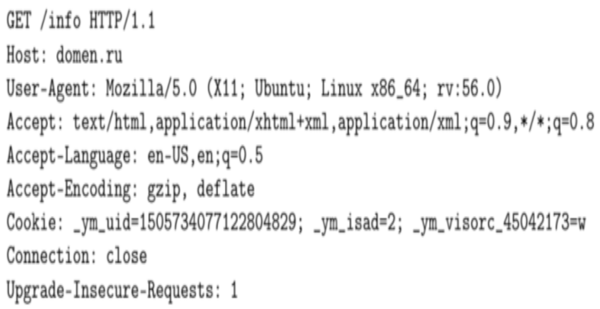
The basic steps to form a model for each query are the following:

- Extracting and analyzing data: analysis of all the incoming requests from the Web browser is carried out.
- Transformation into a vector space model: it is used to transform text data into a vector representation using the TF-IDF algorithm [24], which allows estimating the weight of features for the entire text data array.

Calculation of attribute values: the values of 8 attributes proposed by the author are calculated.

1. Extracting and analyzing data

At the entrance of the Web server requests via HTTP are received. An example of the contents of a GET request is shown in Fig.3.



```
GET /info HTTP/1.1
Host: domen.ru
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:56.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: _ym_uid=1505734077122804829; _ym_isad=2; _ym_visorc_45042173=w
Connection: close
Upgrade-Insecure-Requests: 1
```

Fig. 2. Example of the content fields of HTTP request (GET method)

2. Conversion to a Vector Space Model

To convert strings into a vector form, allowing further application of machine learning methods, an approach based on the TF-IDF method was chosen [24].

TF-IDF is a statistical measure used to assess the importance of words in the context of a document that is part of a document collection or corpus. The weight of a word is proportional to the number of uses of the word in the document and inversely proportional to the frequency of the word use in other documents of the collection. Application of the TF-IDF approach to the problem being solved is carried out for each request.

For each word *t* in the query *d* in the total of queries *D* the value tfidf is calculated according to the following expression:

$$tfidf\,(t,d) = tf\,(t,d)\Box idf\,(t) \qquad (2)$$

The values of *tf, idf* are calculated in accordance with expressions (3), (4) respectively, where *v* is the rest of the words in the query *d*.

$$tf\,(t,d) = \frac{count(t,d)}{\sum\limits_{v \in d} count(v,d)} \qquad (3)$$

$$idf\,(t) = \log \frac{|\,D\,|}{|\,d \in D : t \in d\,|} \qquad (4)$$

Thus, after converting the query *d* ∈ *D* into the vector representation | *d* | it will be set using the set of weights $\{w_t \in T\}$ for each value *t* from the dictionary T.

3. Calculation of attribute values

In [25], 5 basic attributes were proposed for building a detection system computer attacks on web applications:

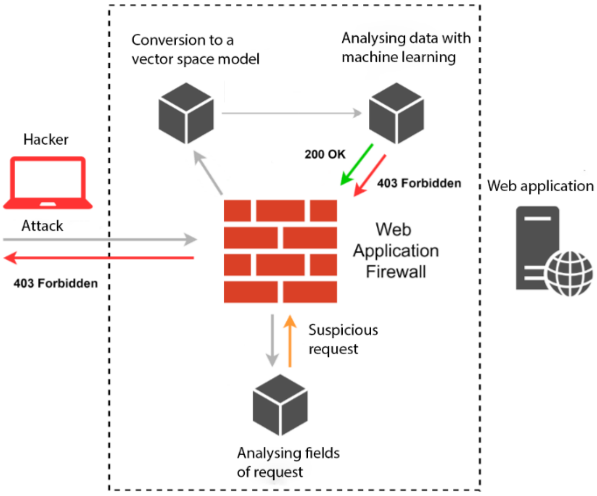


Fig.3 - Analysis of incoming requests for Web applications within the framework of the proposed model

- The length of the request fields sent from the browser (A1).
- The distribution of characters in the request (A2).
- Structural inference (A3).
- Token finder (A4).
- Attribute order (A5).

The author proposed to introduce 3 additional attributes to improve the accuracy of attack detection.

The length of the request sent from the browser (A6)

From the analysis of legitimate requests via the HTTP protocol, it was found out that their length varies slightly. However, in the event of an attack, the length of the data field may change significantly (for example, in the case of SQL injection or cross-site scripting).

Therefore, to estimate the limiting thresholds for changing the length of requests, two of the parameters are evaluated: the expected value μ and variance α^2 for the training set of legitimate data.

Using Chebyshev's inequality, we can estimate the probability that a random variable will take a value far from its mean (expression (5)).

$$P(|\,x - \mu| > \tau) < \frac{\alpha^2}{\tau}\ , \qquad (5)$$

where *x* is a random variable, *τ* is the threshold value of its change.

Accordingly, for any probability distribution with mean μ and variance α^2, it is necessary to choose a value such that a deviation *x* from the mean μ, when the threshold is exceeded, results in blocking the query with the lowest level of errors of the first and second kind.

The attribute value is equal to the probability value from expression (5):

$$A6 = P(|x - \mu| > \tau) \ . \qquad (6)$$

Appearance of new characters (A7)

From the training sample of legitimate requests, we have to select some non-repeating characters (including various encodings) in order to compose the set of symbols of the alphabet A. Thus, when the symbol $b \notin A$ appears in the query, the value of the counter for this attribute is increased by one. The value of the attribute itself is calculated as the ratio of the counter value to the power of the alphabet set:

$$A7 = \frac{p_b}{|A|} \qquad (7)$$

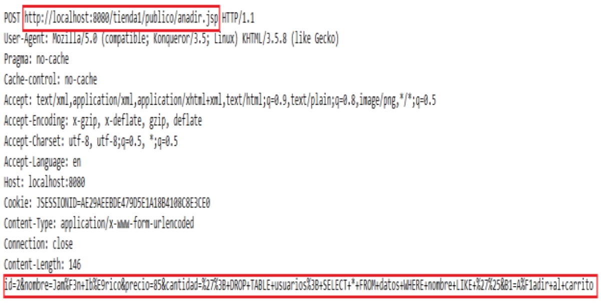
The emergence of new keywords (A8)

From the training sample of legitimate queries, we have to select some non-repeating terms (words) - t in order to compose a set of terms of the dictionary. Thus, when the word $\omega \notin T$ appears in the query, the counter value p_ω for this attribute is increased by one. The value of the attribute itself is calculated as the ratio of the value of the counter to the power of the set of terms of the dictionary:

$$A8 = \frac{p_\omega}{|T|} \qquad (8)$$

IV. CONCLUSION

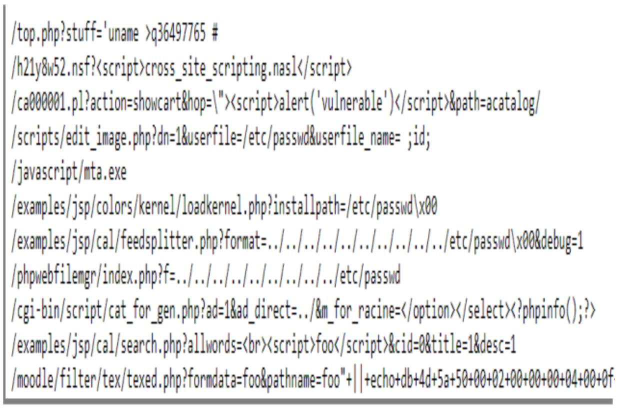
For testing the operation of machine learning methods, a data set from several data sources of system protection tools will be used, such as log files of the intrusion detection and prevention system, HTTP requests (GET, POST method) of the web application firewall, etc.



```
POST http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=AE29AEEBDE479D5E1A18B4108C8E3CE0
Content-Type: application/x-www-form-urlencoded
Connection: close
Content-Length: 146
id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito
```

Fig. 4. An example of the complete dangerous HTTP request with the POST method

When analyzing a full HTTP request, the author focuses on the data in a red frame (Fig. 3). After the extraction process, the data will be saved in the appropriate files (good_request.txt and bad_request.txt). The structure of these files is shown in Fig. 4.



```
/top.php?stuff='uname >q36497765 #
/h21y8w52.nsf?<script>cross_site_scripting.nasl</script>
/ca000001.pl?action=showcart&hop=\"><script>alert('vulnerable')</script>&path=acatalog/
/scripts/edit_image.php?dn=1&userfile=/etc/passwd&userfile_name= ;id;
/javascript/mta.exe
/examples/jsp/colors/kernel/loadkernel.php?installpath=/etc/passwd\x00
/examples/jsp/cal/feedsplitter.php?format=../../../../../../../../../../etc/passwd\x00&debug=1
/phpwebfilemgr/index.php?f=../../../../../../../../../etc/passwd
/cgi-bin/script/cat_for_gen.php?ad=1&ad_direct=../&m_for_racine=</option></select><?phpinfo();?>
/examples/jsp/cal/search.php?allwords=<br><script>foo</script>&cid=0&title=1&desc=1
/moodle/filter/tex/texed.php?formdata=foo&pathname=foo"+||+echo+db+4d+5a+50+00+02+00+00+00+04+00+0f
```

Fig.5. File of dangerous HTTP request

A preliminary study allowed us to obtain an estimate of the accuracy of detecting attacks on Web applications of 96% for the data set [15] using the entered query attributes, query vector representation models and classifier based on decision trees. This fact allows us to conclude that it is possible to build an algorithm for detecting computer attacks on Web applications based on the proposed model for presenting requests to Web resources based on the vector space model and differing in the attribute attributes of requests via HTTP.

REFERENCES

[1]]. Kaspersky Lab. Security report. - 2019. - (дата обращения: 15.04.2019). http:/ / www. securelist. com / en / analysis / 204792244 / The - geography - of - cybercrime - Western - Europe- and-North-America.

[2]. A survey of intrusion detection techniques in cloud / C. Modi [et al.] // Journal of Network and Computer Applications. - Vol. 36, no. 1. - P. 42-57, 2013.

[3]. Khamphakdee N., Benjamas N., Saiyod S. Improving intrusion detection system based on snort rules for network probe attack detection // Information and Communication Technology (IColCT), 2014 2nd International Conference On. - IEEE. - P. 69-74. 2014.

[4]. A stateful intrusion detection system for world-wide web servers / G. Vigna [et al.] // Computer Security Applications Conference, 2003. Proceedings. 19th Annual. - IEEE.. - P. 34-43., 2003

[5]. Sekar R. An Efficient Black-box Technique for Defeating Web Application Attacks. // NDSS. - 2009.

[6]. Mutz D., Vigna G., Kemmerer R. An experience developing an IDS stimulator for the blackbox testing of network intrusion detection systems // Computer Security Applications Conference, 2003. Proceedings. 19th Annual. - IEEE- P. 374-383, . 2003..

[7]. Li X., Xue Y. BLOCK: a black-box approach for detection of state violation attacks towards web applications // Proceedings of the 27th Annual Computer Security Applications Conference. - ACM - P. 247-256, 2011.

[8]. Saxena P., Sekar R., Puranik V. Efficient fine-grained binary instrumentationwith applications to taint-tracking // Proceedings of the 6th annual IEEE/ACM international symposium on Code generation and optimization. - ACM..- P. 74-83, 2008.

[9]. Браницкий А. А., Котенко И. В. Анализ и классификация методов обнаружения сетевых атак // Труды СПИИРАН. - Т. 2, № 45. - С. 207—244, 2016.

[10]. Heckerman D. A tutorial on learning with Bayesian networks // Innovations in Bayesian networks. - Springer. - P. 33-82, 2008.

[11]. Friedman N., Geiger D., Goldszmidt M. Bayesian network classifiers // Machine learning. - - Vol. 29, no. 2-3. - P. 131-163, 1997.

[12]. Goldszmidt M. Bayesian network classifiers // Wiley Encyclopedia of Operations Research and Management Science. - 2010.

[13]. Barbara D., Wu N., Jajodia S. Detecting novel network intrusions using bayes estimators // Proceedings of the 2001 SIAM International Conference on Data Mining. - SIAM. - P. 1-17, . 2001 .

[14]. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы / Ю. Г. Емельянова [и др.] // Программные системы: теория и приложения. - Т. 2, № 3. - С. 3-15., 2011.

[15]. A Detailed Analysis of the KDD CUP 99 Data Set / M. Tavallaee [и др.] // Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications. - Ottawa, Ontario, Canada: IEEE Press. - C. 53—58. - (CISDA’09). - URL: http://dl.acm.org/citation.cfm?id= 1736481.17 36489, 2009.

[16]. Васильев В.И., Шарабыров И.В. Интеллектуальная система обнаружения атак в ло¬кальных беспроводных сетях // Вестник Уфимского государственного авиационного тех¬нического университета. - 2015. - Т. 19, 4 (70).

[17]. Su M.-Y. Real-time anomaly detection systems for Denial-of-Service attacks by weighted k- nearest-neighbor classifiers // Expert Systems with Applications. - Vol. 38, no. 4. - P. 3492-3498. - 2011.

[18]. Lee C. H., Chung J. W., Shin S. W. Network intrusion detection through genetic feature selection // Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on. - IEEE - P. 109-114, 2006.

[19]. Intrusion detection with genetic algorithms and fuzzy logic / E. Ireland [et al.] // UMM CSci senior seminar conference..- Pp. 1-6, 2013.

[20]. Kruegel C., Toth T. Using decision trees to improve signature-based intrusion detection // Recent Advances in Intrusion Detection. - Springer - P. 173-191, 2003.

[21]. Bouzida Y., Cuppens F. Neural networks vs.

ABOUT THE AUTHORS

Manh Thang Nguyen

Workplace: Information Technology Faculty – Academy of cryptography techniques.

Email: chieumatxcova@gmail.com

Training process:

2005-2007: Student at the Military Technical Academy.

2007-2013: Student at the Applied Mathematics and Informatics Faculty - Lipetsk State Pedagogical University – Russia Federation.

2017-present: Post-graduate student at the Military Academy of the Federal Guard Service Russian Federation.

Research today: Computer network, network security, machine learning and data mining.

D.S. Alexander Kozachok

Workplace: The Academy of Federal Guard Service of the Russian Federation.

Email: alex.totrin@gmail.com

The education process: has received PhD. degree in Engineering Sciences in Academy of Federal Guard Service of the Russian Federation in Dec. 2012.

Research today: Information security; Unauthorized access protection; Mathematical cryptography; theoretical problems of computer.