UDC 004.056.57
Doi: 10.31772/2712-8970-2021-22-3-414-424

Для цитирования: Жуков В. Г., Пигалев Я. В. Обнаружение информационного взаимодействия объектов информационной системы с DGA доменами // Сибирский аэрокосмический журнал. 2021. Т. 22, № 3. С. 414–424. Doi: 10.31772/2712-8970-2021-22-3-414-424.

For citation: Zhukov V. G., Pigalev Y. V. Detection of information system objects interaction with DGA domains. *Siberian Aerospace Journal*. 2021, Vol. 22, No. 3, P. 414–424. Doi: 10.31772/2712-8970-2021-22-3-414-424.

Обнаружение информационного взаимодействия объектов информационной системы с DGA доменами

В. Г. Жуков, Я. В. Пигалев*

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
*E-mail: pigalevyan1998@mail.ru

В настоящее время разработчики вредоносного программного обеспечения активно применяют технику генерации доменных имен DGA для установления информационного взаимодействия между вредоносным программным обеспечением и их командными центрами управления. Генерация доменных имен в соответствии с заданным алгоритмом позволяет вредоносному программному обеспечению обходить блокировки средств защиты информации, делая их малоэффективными
и устанавливать канал связи для получения команд управления и их параметров, а также для передачи информации из информационной системы на внешние ресурсы, контролируемые злоумышленниками. Таким образом, необходимо разрабатывать новые подходы к решению задачи обнаружения сгенерированных с помощью DGA доменных имен в DNS трафике информационной системы.

В рамках проведенного исследования авторами разработано решение для обнаружения информационного взаимодействия объектов информационной системы с DGA доменами, основанное
на применении машинного обучения. Обнаружение информационного взаимодействия происходит в два этапа. На первом этапе методами машинного обучения решается задача классификации для каждого DNS имени из общего потока DNS запросов информационной системы. На втором этапе для каждого DNS имени, классифицированного как DGA, осуществляется обогащение данными из внешних источников и принятие окончательного решения о вредоносном характере запроса на разрешение данного DNS имени с последующим оперативным уведомлением администратора безопасности по каналам электронной почты.

В работе приведено описание процесса разработки классификатора на основе машинного обучения, определены входные характеристические данные DNS имени, необходимые для классификации, представлены результаты обучения классификатора на представительном множестве тестовых данных. Обоснована логика принятия решения о вредоносном характере DNS запросов. Разработанное решение было апробировано в рамках экспериментального стенда. Предложены рекомендации по поддержке корректной работы классификатора на основе машинного обучения.

Применение разработанного решения сделает возможным апостериорное обнаружение информационного взаимодействия вредоносного программного обеспечения со скомпрометированных объектов информационной системы с серверами командных центров управления злоумышленников.

Ключевые слова: информационная безопасность, DNS, Domain Generation Algorithm.

Detection of information system objects interaction with DGA domains

V. G. Zhukov, Y. V. Pigalev*

Reshetnev Siberian State University of Science and Technology
31, Krasnoyarsky rabochy Ave., Krasnoyarsk, 660037, Russian Federation
*E-mail: pigalevyan1998@mail.ru

Currently, malware developers are actively using domain name generation technique called DGA to establish communication between malware and its command centers. Domain name generation in accordance with the given algorithm allows malicious software to bypass information protection tools blacklists, thus making blacklists ineffective, and establish a communication channel to receive control commands and parameters, as well as to transfer information from the information system to external resources controlled by attackers. Thus, it is necessary to develop new approaches to DGA generated domain names detection using DNS traffic of an information system.

During the research, the authors have developed a solution for detecting information objects interaction with DGA domains based on the use of machine learning. The detection of this interaction occurs in two stages. On the first stage the classification task is being solved for each DNS name from overall information system DNS stream. On the second stage, for each DNS name classified as DGA, corresponding DNS query is being enriched using data from external sources and a final decision about the malicious nature of the query to resolve this DNS name is being made, followed by a notification of a security administrator via e-mail channels.

The paper describes the process of developing a classifier based on machine learning, defines the input data of the DNS name necessary for classification, presents the results of classifier training on a representative set of test data. The logic of making a decision about the malicious nature of DNS queries has been substantiated. The developed solution was tested using an experimental stand. Some recommendations for correct classifier operation support are proposed.

The application of the developed solution will make possible posteriori detection of information interaction of malicious software working on compromised information objects with the servers of attackers command and control centers.

Keywords: information security, DNS, Domain Generation Algorithm.

Introduction

The DNS protocol is infrastructure-forming and, as a rule, is allowed by default in the information systems of organizations, regardless of their sphere of activity. DNS traffic information flows, in general, are either insufficiently controlled or not controlled at all. It is for this reason that modern malicious software (malware) frequently uses the DNS protocol to communicate with control servers (C&C, Command and Control Server), which is confirmed by numerous studies, for example, Spamhaus for 2019 [1].

Information security tools prevent this interaction by detecting and blocking DNS queries for resolving domain names of C&C centers, for example, using a blacklist mechanism. To circumvent these restrictions, attackers use special software in order to generate domain names in accordance with the given algorithm – Domain Generation Algorithm (DGA). The use of the DGA allows cybercriminals to escape from the static list of C&C domain names and make blacklists used by security tools ineffective; the DGA allows generating an arbitrary number of malicious domains, it is impossible to add them all to the blacklist [2]. Thus, traditional information security tools using black lists are not effective, and a different approach is needed to solve the problem of discovering DGA domains, because the very fact of an outgoing DNS query to resolve the DGA name of the C&C indicates a compromised node within the infrastructure being protected or an attempt at such a compromise. One of the promising solutions is using machine learning methods for automated detection of information interaction of information system objects with DGA domains. As part of the study, the authors have developed an algorithm and software that makes it possible to detect the facts of such information interaction.

DGA Detection Key Stages

A domain network running Microsoft Windows is considered to be the infrastructure of an information system; an information object is understood as any active network node that can generate DNS queries. The interaction of the information object with the DGA domain consists, at least, in the initiation of the DNS object by the information object for the resolution of the DGA domain name.

It is possible to detect the interaction of information objects with DGA domains by the posteriori analysis of DNS query log records.

Local records about DNS queries are forwarded to the domain controller by means of Windows Log Forwarding, where they are further processed to detect DGA domains. A conceptual scheme for detecting information interaction of information objects with DGA domains is shown in Fig. 1.

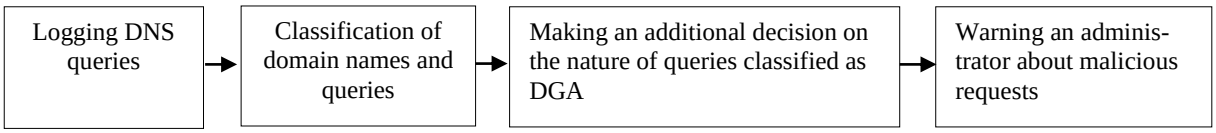

Рис. 1. Схема обнаружения DGA

Fig. 1. DGA detection scheme

Thus, the detection process is divided into two key stages:

1) classification of DNS queries based on machine learning;

2) additional processing of domain names classified as DGA, with the final decision on the malicious nature of the request.

Let us consider the listed stages of the work in more detail.

Stage 1: Classification. All DNS queriers are processed and stored as SQLite database table records on the domain controller. The structure of the table of records is presented in Table. 1.

In the first step, the domain name from each record in the table is classified using machine learning based on the attributes of its domain name. The classification based on domain name attributes was chosen primarily because it is independent of changes introduced by intruders into the malware DGA algorithm [3].

Table 1

Description of DNS table record fields

Record field	Description
ID	Record identifier
query	Domain name
answer	Respond to query (IP address)
time	Time and date of a query
hostname	The name of a network node
status	Query status
image	Application making a query
class	Name class set after processing by the classifier to the value "DGA" or "REAL"

The development of a classifier based on machine learning

The task of the classifier is to assign each domain name to one of two classes:

1) DGA – DNS query to resolve such a name is considered to be malicious;

2) real domain name – DNS query to resolve such a name is considered to be legitimate.

The classifier was developed in Python. A set of scikit-learn libraries [4] were used for machine learning, data processing and classifier evaluation.

According to the analysis report [5–9], the following attributes were selected as the attributes of the domain name, on the basis of which the classification was to be made:

1) domain name length;

2) the ratio of the sum of the lengths of all meaningful words (words found in the dictionaries of the human language) in the domain name to the total name length;

3) the ratio of the length of the longest meaningful word in the domain name to the total name length;

4) the ratio of the number of digits in the domain name to its total length; it is calculated by a formula;

5) Levenshtein distance between the current and the previous domain name – the minimum number of characters that need to be added, removed or changed in order to get the current one from the previous domain name (for example, the Levenshtein distance between test.ru and 1t3st.su is 3). This metric is the most suitable, since unlike, for example, Hamming distance, it does not require the same length of two lines. Moreover, this metric is used in similar studies of the DGA [5];

6) information entropy according to Shannon's definition;

7) the ratio of the number of vowels to the number of consonants of the domain name.

As the classification problem is binary in nature, DGA sampling of generated and real domain names is required. Sampling of names was used for training and testing the classifier, as well as for its final assessment.

Real domain names were taken from the list of the most popular domain names made by DomCop [10], the source of DGA domains is Bambenek Consulting [11] – these sources have already been used in the development of tools for identifying DGA domains [5; 6].

Both samples were 25.000 domain names for a total of 50,000 domain names. The total sample of 50,000 domain names was split into two parts: 80 % was a training sample, the remaining 20 % were a test sample.

The Random Forest algorithm was chosen as a kernel for the classifier, which proved itself positively in solving similar problems [3; 12].

Before training the classifier based on the Random Forest algorithm, preliminary testing of this algorithm was performed using a stratified cross-validation variant with 10 blocks on the training set. Using the cross-validation method, the training sample of domain names is randomly divided into ten blocks of the same size. In turn, each block is considered as a test sample, and the remaining nine blocks are considered as a training sample. For each block, a contingency table is calculated. Further the final contingency table is calculated, averaged over 10 blocks.

The final contingency table (Table 2) presents the values of the average number of correctly defined names, errors of first and second kind as a percentage to the number of domain names in one block consisting of 4000 domain names.

Table 2

Average contingency table for Random Forest when testing by using cross-validation

		Classified, %		Total, %
		DGA	Real	
In fact, %	DGA	48.6	1.26	49.86
	Real	0.77	49.37	50.14
Total, %		49.37	50.63	100

The error rates for the Random Forest algorithm when testing by using cross-validation on a training set are satisfactory.

Based on the final contingency table, the accuracy of the classification was calculated using the following formula:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

where *TP* is the number of true-positive cases; *TN* is the number of true-negative cases; *FP* is the number of false-positive cases; *FN* is the number of false-negative cases.

Random Forest showed high accuracy (98%) when being tested on a training set using cross-validation.

For the final assessment of the trained classifier, a contingency table was calculated. The table was obtained using the classification of the names from a test sample (10,000 domain names). The accuracy was calculated as well.

The contingency table is presented in table. 3.

Table 3

Random Forest contingency table on the test set

		Classified, %		Total, %
		DGA	Real	
In fact, %	DGA	49.42	1.14	50.56
	Real	0.77	48.67	49.44
Total, %		50.19	49.81	100

Random Forest on the test set has the accuracy of 98.09 %.

The results obtained (accuracy, number of the errors of first and second kind) allow us to proceed to the second stage of the study.

Stage 2: Enriching classification results and decision making.

To reduce possible classification errors, all the records in the DNS table of the database (in which domain names are classified as 'DGA') are selected for further enrichment and a decision on their malicious nature.

The decision about the malicious nature of a query is based on the Threat Index calculation, which is calculated based on the parameters of the corresponding DNS query.

The parameters are calculated based on the results of enrichment of a query from external sources of information. They reflect such distinctive properties of DNS queries for DGA name resolution as:

1) one domain name can be resolved to several IP addresses and, according to the EXPOSURE study: Finding Malicious Domains Using Passive DNS Analysis [13], malicious domains of the same malware family are usually resolved to IP addresses of different countries;

2) as a general matter, DGA domains are generated one hour before the attack and are valid within 24 hours [12; 14];

3) DGA names are poorly documented: it is impossible to obtain information about the organization that owns the DGA name, a domain administrator;

4) while malware is running using DGA, malware at the object of the information system goes through a set of generated names in order to find what is available by requesting for each of them. Most of the queries end with the error message 'NXDOMAIN' (nonexistent domain), which indicates that the domain name was not found [12].

The formula for calculating the Threat coefficient is presented below:

$$Threat = \sum_{i=1}^{7} x_i, \quad (2)$$

where x_1 is set to 1 if the number of countries that own IP addresses in responses to a DNS query is more than 2; x_2 is set to 1 if the whois-response to the domain name does not contain the name of the organization of the domain name owner; x_3 is set to 1 if there is no administrator name in the whois-response; x_4 is set to 1 if it was found that the number of DNS queries resulting from the NXDOMAIN error for the current day is greater than the threshold value;

x_5 is set to 1 if the difference between the domain registration date and the DNS query time is less than 1 hour; x_6 is set to 1 if the difference between the domain registration expiration date and the query time is less than 1 day; x_7 is set to 1 if the whois-response does not contain the domain registration date and domain registration expiration date.

The parameters are binary in nature, by default each parameter is 0.

If the Threat Threat Ratio exceeds 3, then a decision is made that the corresponding DNS query is indeed malicious. Otherwise, a decision is made on the legitimacy of the DNS query: a classification error occurrs.

After making a decision, the query, the corresponding decision and the data obtained by enrichment are written to the Suspicious database table to analyze the operation of the tool.

The structure of records in the Suspicious table is shown in Table. 4.

Table 4

Description of additional fields of Suspicious records

Record field	Description
country_number	Number of countries that own IP addresses in response to a query
registrar	Domain administrator name
creation_date	Domain registration date
expiration_date	Registration expiration date
organisation	Name of the organization that owns the domain name

NXDOMAIN_query_count	Number of NXDOMAIN error responses for the combination of an application and computer
domainStatus	Domain name status, 'Up' – available, 'Down' - unavailable
queryType	Decision made on the basis of the calculated parameters about the malicious nature of the DNS query, 'Malicious' - harmful, 'Benign' – legitimate
parentRecord	Link to the corresponding record in the DNS table

Alerts about DNS queries classified and confirmed as DGA are sent to an administrator by email. The alert contains basic information about the corresponding malicious DNS query.

Testing the operation of the detection tool for the interaction of information objects with DGA domains

Testing was conducted on a test Windows domain network consisting of two computers making DNS queries and a domain controller running a DGA detector.

All DNS queries for computers on the domain network were recorded to the domain controller. For testing, DNS queries were made from the computers of the test network to resolve a set of real names and three queries for DGA names.

The queries for DGA names simulated malware enumerating the set of DGA names on the computer in order to find a valid one. For this reason, the first two queries returned an NXDOMAIN error (ijoratsdxgwubk.ru and bsqncknwntpill.ru), the latter returned the IP address of the C&C server (oqunedkxrrrd.ru).

The queries were recorded by the tool on the domain controller from a log file to the DNS table in the database. A fragment of the queries is shown in Fig. 2, the queries for DGA names are highlighted.

ID	query	answer	time	hostname	status	image	class
...	Фильтр	Фильтр	Фильтр	Фильтр	Ф...	Фильтр	Фи...
517	oqunedkxrrrd.ru	::ffff:23.61.215.146	2020-06-13 11:56:03.2...	win8hostClone...	0	C:\virusexample.exe	NULL
518	bsqncknwntpill.ru		2020-06-13 11:55:49.5...	win8hostClone...	9003	C:\virusexample.exe	NULL
519	ijoratsdxgwubk.ru		2020-06-13 11:55:18.3...	win8hostClone...	9003	C:\virusexample.exe	NULL
534	ndj6iayz7u2mbga4pqu4z...	::ffff:185.15.175.157	2020-06-13 11:54:23.4...	win8hostClone...	0	C:\Program Files (x86)\Internet ...	NULL
535	ndj6iayz7u2mbga4pqu4z...	::ffff:185.15.175.158	2020-06-13 11:54:23.4...	win8hostClone...	0	C:\Program Files (x86)\Internet ...	NULL

Рис. 2. Выгруженные DNS запросы

Fig. 2. Stored DNS quieries

On completing recording the queries, the classification stage began: all domain names from the database were classified based on machine learning. Further the second stage began: all the records from the table, resulting in the DGA classification, were selected to enrich and determine the nature of queries.

The queries were enriched using whois-requests for the domain name, geolocation checking the IP address, counting the number of NXDOMAIN responses, then the parameters were calculated based on the enrichment results.

For the combination of application and computer corresponding to these test DGA queries ("win8hostClone" and "C: \ virusexample.exe"), there were 2 domain queries in the database with the NXDOMAIN error, which caused the x_4 parameter to be set to 1 for each record with the corresponding combination of application and computer name, all three domains are unavailable (thus, there is no information about the organization, domain administrator, registration start and end dates, which set the appropriate parameters x_2, x_3, x_7 to 1). The values of the other parameters remained by default.

Therefore, according to the formula (2), the threat ratio for test DGA queries was is equal to:

$$Threat = 0 + 1 + 1 + 1 + 0 + 0 + 1 = 4. \quad (3)$$

The threat factor value is 4, on the basis of which it was decided that the queries were malicious as a matter of fact.

After classification, enrichment, character determination, the selected records with additional information were recorded into the Suspicious database table.

The fragment of the Suspicious table with the data obtained during enrichment for the queries shown in Fig. 2 is shown in Fig. 3. For all the three malicious test requests, a decision was made about their malicious nature (this is indicated by the value of "Malicious" in the field "queryType"), for real domain queries, a decision was made about their legitimacy (this is indicated by the value of "Benign" in the "queryType" field).

ID	query	answer	time	hostname	image	country_number	registrar	queryType
119	oqunedkxrrrd.ru	::ffff:23.61.215.1...	2020-06-13 ...	win8hostClone.d...	C:\virusexample.exe	1	Error, domain is unavaible	Malicious
120	bsqnckrwntpill.ru		2020-06-13 ...	win8hostClone.d...	C:\virusexample.exe	0	Error, domain is unavaible	Malicious
121	ijoratsdxgwubk.ru		2020-06-13 ...	win8hostClone.d...	C:\virusexample.exe	0	Error, domain is unavaible	Malicious
122	ndj6iayz7u2mbg...	::ffff:185.15.175....	2020-06-13 ...	win8hostClone.d...	C:\Program Files (x...	1	RU-CENTER-RU	Benign
123	ndj6iayz7u2mbg...	::ffff:185.15.175....	2020-06-13 ...	win8hostClone.d...	C:\Program Files (x...	1	RU-CENTER-RU	Benign
124	t3848077496801...	::ffff:50.116.239....	2020-06-13 ...	win8hostClone.d...	C:\Program Files (x...	1	Amazon Registrar, Inc.	Benign
125	ndj6iayz7u2mbg...	::ffff:185.15.175....	2020-06-13 ...	win8hostClone.d...	C:\Program Files (x...	1	RU-CENTER-RU	Benign
126	ndj6iayz7u2mbg...	::ffff:185.15.175....	2020-06-13 ...	win8hostClone.d...	C:\Program Files (x...	1	RU-CENTER-RU	Benign

Рис. 3. Результат работы средства

Fig. 3. Results

The administrator was notified of every malicious query for a DGA name. The test notification about one of the malicious queries is shown in Fig. 4.

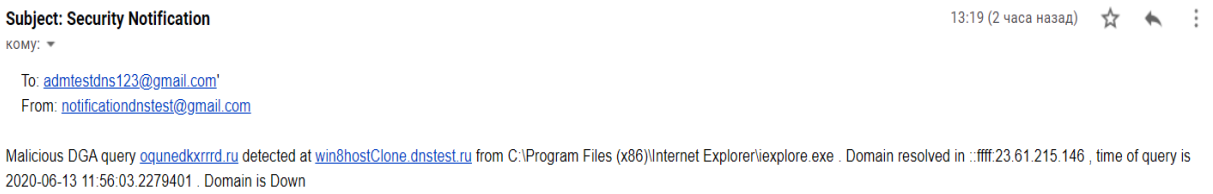

Рис. 4. Оповещение об обнаружении вредоносного запроса

Fig. 4. Notification about a detected malicious quiery

Thus, during testing, the interaction of information objects with DGA domains was found. The data on malicious queries, being stored in the database and sent by e-mail to the administrator, make it possible to determine the fact and circumstances of the compromise of an information object.

Support for the correct operation of the classifier based on machine learning

Classifiers based on machine learning algorithms degrade over time. What is worse, classifiers tend to have lower performance in practical conditions than they did in testing [15].

It is assumed that, in addition to unpredictable changes, the accuracy of a classifier may decrease over time due to changes in the general DGA algorithm for generating malicious domains (i.e., the name structure will change).

Thus, to maintain the accuracy of a classifier, it is necessary to monitor its operation and, if necessary, modify it and / or retrain: change the set of domain name attributes, train the classifier on a newer set of domain names.

Conclusion

Based on the results of the research, the authors developed and programmatically implemented a two-stage DGA detection algorithm: classification using machine learning based on the Random Forest algorithm and deciding on the nature of queries based on the enrichment results.

Using the developed software allows posteriori detection of the interaction of information objects with DGA domains. Thus, it becomes possible to detect the fact that an information object is compromised and to increase its security by jointly using the developed tool with other information security systems.

The detection tool is designed to analyze DNS queries on a Microsoft Windows domain network, but its core, which is a machine learning classifier and malware decision logic, can be applied to other operating systems and hardware as well.

Библиографические ссылки

1. Spamhaus Botnet Threat Report 2019 [Электронный ресурс]. URL: https://www. spamhaus.org/news/article/793/spamhaus-botnet-threat-report-2019 (дата обращения 02.02.2020).

2. Threat Brief: Understanding Domain Generation Algorithms (DGA) [Электронный ресурс]. URL: https://unit42.paloaltonetworks.com/threat-brief-understanding-domain-generation-algorithms-dga/ (дата обращения 05.08.2020).

3. Sivaguru R., Choudhary C. An Evaluation of DGA Classifiers // IEEE International conference on Big Data, Seattle, USA, 2018. P. 5058–5067.

4. Scikit-learn: machine learning in Python [Электронный ресурс]. URL: https://scikit-learn.org/stable (дата обращения 03.01.2020).

5. Li Y., Xiong K. Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection. IEEE Access, 2019. P. 32765–32782.

6. Anderson H. S., Woodbridge J. DeepDGA: Adversarially – Tuned Domain Generation and Detection. Proceedings of the 2016 ACM Workshop and Artificial Intelligence and Security. 2016. P. 13–21.

7. Anderson H. S., Woodbridge J. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. Endgame, Inc, 2016. 13 p.

8. Gupta B., Sheng M. Machine Learning for Computer and Cyber Security: Principles, Algorithms, and Practices. Taylor and Francis Group, 2019. 364 p.

9. Alazab M., Tang M. Deep Learning Applications for Cyber Security. Springer Nature Switzerland, 2019. 246 p.

10. Top 10 million Websites based on Open data from Common Crawl & Common Search [Электронный ресурс]. URL: https://www.domcop.com/top-10-million-websites (дата обращения 03.02.2020).

11. Bambenek Consulting [Электронный ресурс]. URL: http://osint.bambenekconsulting.com/feeds/dga-feed.txt (дата обращения 16.01.2020).

12. Wang Z., Jia Z. A Detection Scheme for DGA Domain Names. SVM Proceedings of the 2018 International Conference on Mathematics, Modelling, Simulation and Algorithms. New York, USA, 2018. P. 257–263.

13. Bilge L., Kirda E. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. Proceedings of the Network and Distributed System Security Symposium, San Diego, USA, 2011. 17 p.

14. Plohmann D., Yakdan K. A Comprehensive Measurement Study of Domain Generating Malware. Proceedings of the 25th USENIX Security Symposium, Austin, USA, 2016. P. 263–278.

15. Why Machine Learning Models Degrade in Production [Электронный ресурс]. URL: https://towardsdatascience.com/why-machine-learning-models-degrade-in-production-d0f2108e9214 (дата обращения 25.05.2020).

References

1. Spamhaus Botnet Threat Report 2019. Available at: https://www.spamhaus.org/news/article/793/spamhaus-botnet-threat-report-2019 (accessed: 02.02.2020).

2. Threat Brief: Understanding Domain Generation Algorithms (DGA). Available at: https://unit42.paloaltonetworks.com/threat-brief-understanding-domain-generation-algorithms-dga/ (accessed: 05.08.2020).

3. Sivaguru R., Choudhary C. An Evaluation of DGA Classifiers. IEEE International conference on Big Data, Seattle, USA, 2018, P. 5058–5067.

4. Scikit-learn: machine learning in Python. Available at: https://scikit-learn.org/stable (accessed: 03.01.2020).

5. Li Y., Xiong K. Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection. IEEE Access, 2019, P. 32765–32782.

6. Anderson H. S., Woodbridge J. DeepDGA: Adversarially – Tuned Domain Generation and Detection. Proceedings of the 2016 ACM Workshop and Artificial Intelligence and Security, 2016, P. 13–21.

7. Anderson H. S., Woodbridge J. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. Endgame, Inc, 2016, 13 p.

8. Gupta B., Sheng M. Machine Learning for Computer and Cyber Security: Principles, Algorithms, and Practices. Taylor and Francis Group, 2019, 364 p.

9. Alazab M., Tang M. Deep Learning Applications for Cyber Security. Springer Nature Switzerland, 2019, 246 p.

10. Top 10 million Websites based on Open data from Common Crawl & Common Search. Available at: https://www.domcop.com/top-10-million-websites (accessed 03.02.2020).

11. Bambenek Consulting. Available at: http://osint.bambenekconsulting.com/feeds/dga-feed.txt (accessed 16.01.2020).

12. Wang Z., Jia Z. A Detection Scheme for DGA Domain Names. SVM Proceedings of the 2018 International Conference on Mathematics, Modelling, Simulation and Algorithms, New York, USA, 2018, P. 257–263.

13. Bilge L., Kirda E. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. Proceedings of the Network and Distributed System Security Symposium, San Diego, USA, 2011, 17 p.

14. Plohmann D., Yakdan K. A Comprehensive Measurement Study of Domain Generating Malware. Proceedings of the 25th USENIX Security Symposium, Austin, USA, 2016, P. 263–278.

15. Why Machine Learning Models Degrade in Production. Available at: https://towardsdatascience.com/why-machine-learning-models-degrade-in-production-d0f2108e9214 (accessed 25.05.2020).

Жуков Вадим Геннадьевич – кандидат технических наук, доцент кафедры безопасности информационных технологий; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mai: zhukov@mail.sibsau.ru.

Пигалев Ян Вячеславович – магистрант; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: pigalevyan1998@mail.ru.

Zhukov Vadim Gennadevich – Cand. Sc., Associate Professor at the Department of Information Technology Security, Reshetnev Siberian State University of Science and Technology. E-mail: zhukov.sibsau@gmail.com.

Pigalev Yan Vyacheslavovich – Master's Degree Student; Reshetnev Siberian State University of Science and Technology. E-mail: pigalevyan1998@mail.ru.