DNS-Based Fast-Flux Botnet Detection Approach

Sergii Lysenko[a], Kira Bobrovnikova[a], Piotr Gaj[b], and Oleg Savenko [a]

[a] *Khmelnitsky National University, Khmelnitsky, Ukraine*
[b] *Silesian University of Technology, Gliwice, Poland*

Abstract
Today the problem of botnets detection is very actual, as botnet are widespread and are used to perform different types of cyberattacks and to cause threats to network services and users' properties. One of the mean the botnets use to connect with their command-and-control (C&C) is the domain name system (DNS). On other hand the fast-flux technique enables to avoid botnets' detection. The pa-per presents a new botnets' detection technique, which takes into account the DNS feature analysis, botnets' architecture aspects, as well as theirs behaviors in the network and hosts. Proposed approach allows detecting the botnets' bots of centralized, decentralized and hybrid architecture with high efficiency.

Keywords 1
Bot, botnet, detection, fast-flux, DNS, DNS traffic, network, botnet, detection, network, malware

1. Introduction

During recent years, such a phenomenon as a botnet has been one of the most dangerous types of malwares. Botnets are a powerful tool for cybercrime, such as DDoS attacks, banking fraud, cyber espionage, and malware distribution, usage of the IoT compromised devices and are used to organize anonymous proxies, provide remote machine service, as well as spread spam, click fraud, phishing, etc. These criminal acts cause significant harm to both individual users and the global economy as a whole [1-3].

The vast majority of botnets use the Domain Name System (DNS) to control the infected computer systems [4-7]. Using the DNS service gives an attacker the ability to anonymously and flexibly manage the botnet and increases its reliability. In order to develop, control, maintain, and conceal the infrastructure of the botnets' com-mand-and-control (C&C), cybercriminals use a variety of techniques, including DNS-based evasion technologies. One of the most difficult to detect and actively used by botnet masters evasion technique is fast-flux. The detection of the botnet and the countermeasures implementation in order to terminate their activities is complicated by the possibility of anonymous management and dynamic geographically distributed structure of botnets. Thus, given the growing number of cybercrimes committed using botnets [8-10], the urgent task is to construct the new approaches for the detection of botnets to boost the detection efficiency.

2. Related work

The state-of-art concerning the botnet detection techniques is presented in this section. Thus, in [11] an approach for botnet detection is presented. It enables the procedure of scaling for every part of the known botnets, that makes it possible to per-form effective botnet detection. For this purpose, the researchers involved such methods as machine learning and statistical approaches. The paper

ICTERI-2021, Vol II: Workshops, September 28 – October 2, 2021, Kherson, Ukraine
EMAIL: sirogyk@ukr.net; bobrovnikova.kira@gmail.com; piotr.gaj@polsl.pl; savenko_oleg_st@ukr.net
ORCID: 0000-0001-7243-8747; 0000-0002-1046-893X; 0000-0002-2291-7341; 0000-0002-4104-745X


concluded that usage of the proposed technique demonstrated that the approach had good results for some types of botnets.

In [12] the principles of the botnets functioning as well as damage they cause are presented. Paper also presents the set of botnet detection techniques in cyber-physical systems.

The survey devoted to botnet detection problem is presented in [13] An article presents the botnets techniques based on theirs functioning features and architectures. In addition, authors have overviewed the set of approaches for fast-flux botnets host-, router- and DNS-based approaches for the efficient detection.

In article [14] the principles of the fast-flux botnet spread in the networks are presented. The paper also provides an approach for the fast-flux botnets detection in the Internet Service Provider network infrastructures. For this purpose, authors gathered high amount of DNS traffic and make decision about botnet presence using the K-means clustering.

An analysis of the Rustock botnet is presented in [15]. Paper is devoted to the fast-flux detection problem and present the network traffic features, that may indicate the botnet presence.

In [16] a DNS-based approach for the detection of the botnet was presented. It employs the DNS query and response behaviors analysis, and is able to detect bot-nets with good efficiency.

An article [17] provides the survey of the DNS-based detection techniques. It enables botnet detection via deep analysis of the DNS traffic.

The paper [18] presents an approach for "zero-day" online fast-flux botnet detection. It uses a new adaptive approach for the detection. It is based on the new algorithm, trained with the usage of fuzzy neural network. It makes it possible to perform the traffic classification, and it demonstrates good results. The proposed framework is based on the investigation of the set of features of the fast-flux networks and uses DNS traffic database. A new framework for fast-flux botnet detection called Fast Flux Killer System (FFKS) is presented in [19]. It employed an adaptive dynamic evolving spiking neural network algorithm for the network traffic classification.

In [20] an approach that is able to recognize the presence via analysis of the group of botnet's domains. Such group may be formed by a special domain generator. Also, approach analyzes possible variants of the botnet's domain groups that belong to other botnets. Technique is able to detect if the domain names group was formed into a specified botnet using the fast-flux technique. For this purpose, the approach uses the mechanism known as a double-stages detection. In [21] an approach for fast-flux botnet detection based on the passive analysis of the DNS traffic is pro-posed. It involves the analysis of the set of features, that may indicate botnet presence. The proposed technique is based on the near-real-time identification of different metrics that measure a wide range of botnets' fast-flux features. Author propose to combined needed metrics via a mathematical and data mining approach. The paper [22] presents a new framework called "The Gunner System". It is filtering approach, that involves the rule-based DNS features of the botnets detection.

A new system for the detection of botnets called BotGRABBER was presented in [23,24]. It serves as an intrusion detection system, has an adaptive nature, and makes it possible to provide resilient functioning of the network infrastructure in the situation of the cyberattacks caused by botnets. While attacks are being performed, BotGRABBER produces needed security scenarios according to the network state for the attack's mitigation.

In [25,26] the techniques for botnets malware detection are presented. As Intrusion Detection System it performs the botnet's detection with good effectiveness.

The state-of-art showed that great variety of the techniques have been proposed and they demonstrate sufficient detection. However, the false positives presence attests to the fact that proposed approaches do not combine detection via botnets' features analysis, architectural aspects analysis, and botnet's behavior analysis on theirs different lifecycle stages.

3. DNS-Based Fast-Flux Botnet Detection Technique

3.1. Fast-Flux Botnet Lifecycle

In this section we present a new approach for DNS-based fast-flux botnet detection, which is based on botnets' models creation. Proposed models describe the botnets' functioning and take into

consideration as features analysis, as architectural particularities, as well as botnet's behavior in the hosts presented in computer networks.

To develop a new botnet's functioning model, let us consider the its life-cycle. In general, each typical botnet has the set of stages: (1) infection of the host or penetration into the host; (2) attempt to register in the host and attempt to connect to the command and control center called as C&C server; (3) attempt to execute the malicious actions in the network or host; (4) maintenance; (5) attempt to execute the self-destruction [27].

Unlike uninfected computer systems on the local network, which typically use local DNS servers for DNS queries, infected with computer systems can also use free DNS services (OpenDNS, FreeDNS) or their own DNS servers. A scheme of the establishing a connection between the botnets and the C&C server using DNS is presented on Fig.1. The queries and responses sequence are represented by the notation from 1 to 10. DNS queries are initiated by the bot in the following situations: (1) for the initial registration of the bot and integration with the botnet after successful computer system infection; (2) after the connection to C&C server fails (after the 3-stage is failed, the "handshake" the bots start sending requests to the DNS server); (3) after migrating of the C&C server; (4) after changing the IP address of the C&C server; (5) when executing the malicious actions (DDoS-attacks, spam-mailings etc.); (6) after rebooting of the infected computer system; (7) in order to increase noise immunity (to obtain additional domain names group, that is related to the botnets operation, bot performs the reverse DNS queries).

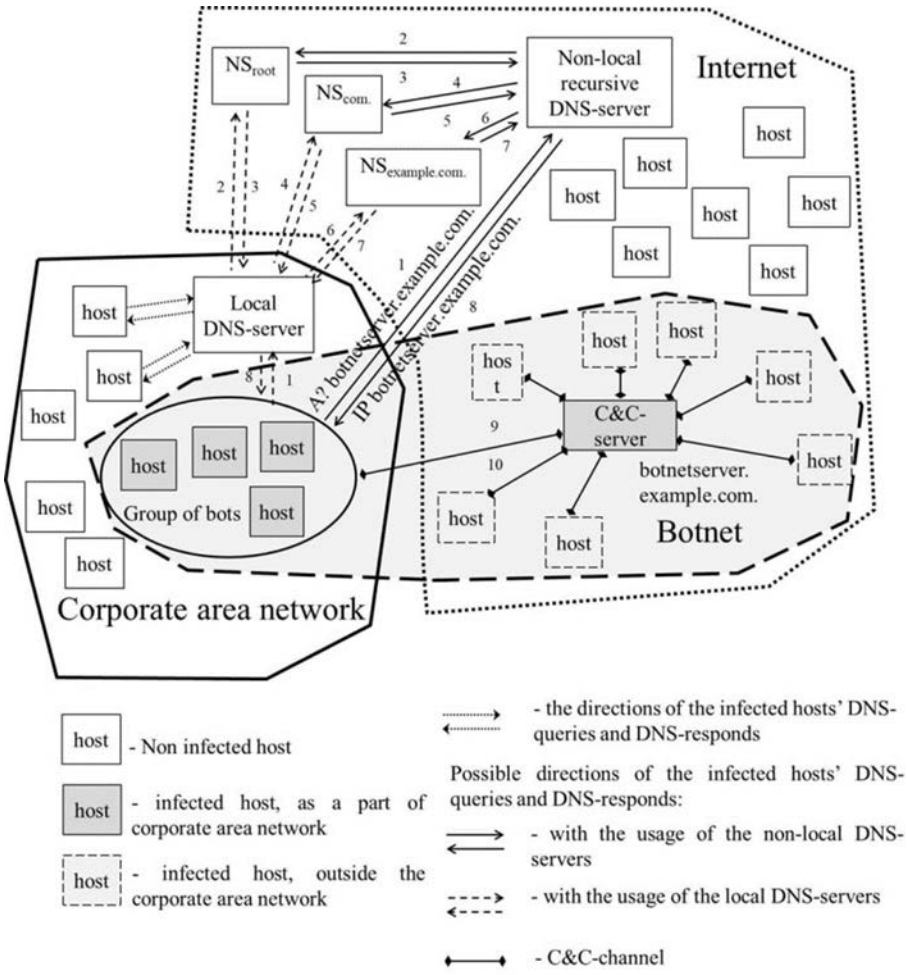

Figure 1: The scheme of the establishing the connection between botnets and C&C server using DNS

One of the characteristic feature DNS-queries from the infected computer system is the group activity [28]. The botnets' bots perform the simultaneous or concentrated in a short period DNS-queries when attempting to connect to C&C-servers, to perform procedure of migration, to execute the actions or to download the malware updates.

In situation, when the DNS resource records concerning the domain name were cached by the DNS client of the computer system, the repeated DNS request does not go beyond the local DNS cache of the computer system before the TTL is expired. A lot of botnets ignore the duration of TTL period presented in response from the authoritative DNS server to the DNS request. It means that the component of botnet tries to clear the local DNS cache and resends the domain name before the end of the TTL period. It allows bot master increasing the flexibility and reliability of bot management Fig. 2, a, b, in the sequence of actions is represented by the notation from 1 to 6).

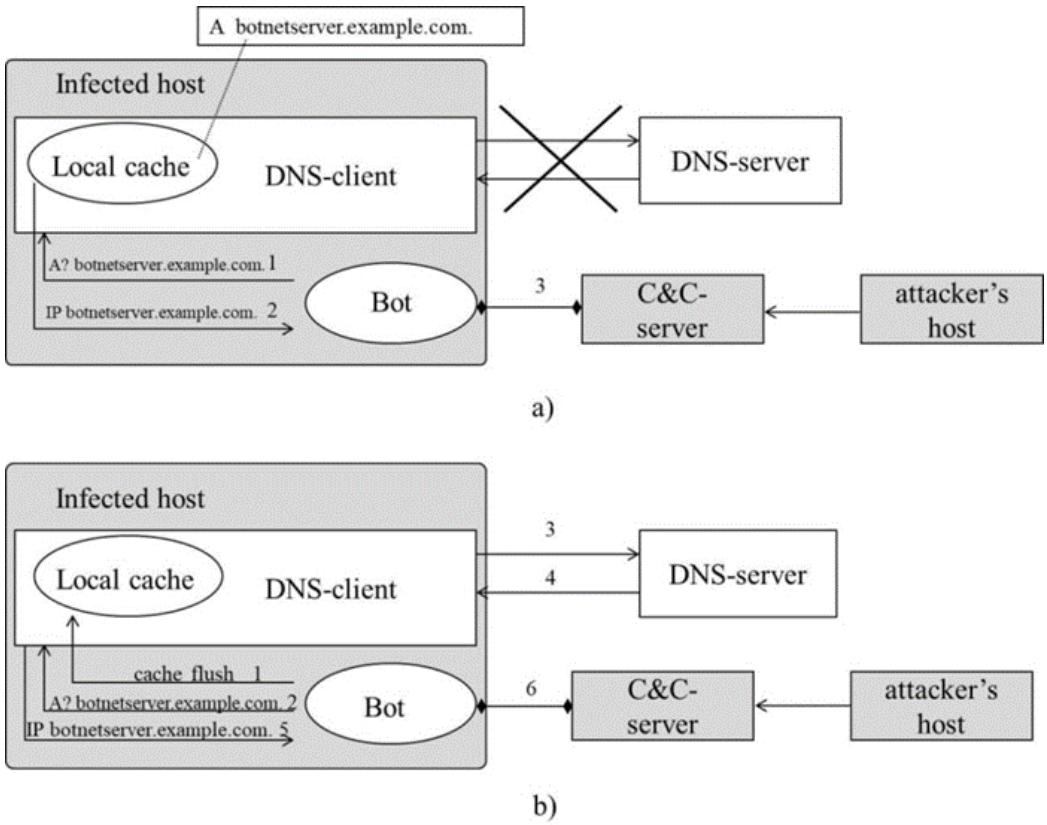

Figure 2: A repeated DNS request for a domain name before the end of the TTL period: a) in case of a cached DNS response in the local DNS cache; b) after clearing the local DNS cache

3.2. DNS-Based Fast-Flux Botnet Model

Let us define the DNS-based fast-flux botnet model in the point of view of a bot's management during its lifecycle. Let present is as a:

$$M_{BN} = \langle C, A, B, Z, L, F \rangle, \quad (1)$$

where $C = \{c_j\}_{j=1}^{N_N} = \{\langle\langle D, I\rangle, \langle N, E\rangle\rangle\}_{j=1}^{N_C}$, $D = \{d_j\}_{j=1}^{N_D}$, $I = \{i_j\}_{j=1}^{N_I}$, $N = \{n_j\}_{j=1}^{N_N}$, $E = \{e_j\}_{j=1}^{N_E}$ – the set of domain names of the botnet's control elements, theirs appropriate IP-addresses; the set of the domain names and IP-addresses of the authoritative servers for names d respectively, N_D – C&C domain names number, N_I – number of IP-addresses of the C&C domain names, N_N - number of authoritative servers domain names, N_E – number of authoritative servers' IP addresses; N_C – a number of the controlled elements of the botnet; $A = \{a_j\}_{j=1}^{3}$ – botnet architecture type, a_1 – centralized, a_2 – decentralized, a_3 –hybrid (Fig.3); $B = \{b_j^p\}_{j=1}^{N_B}$ – set of network protocols, employed for botnets management in the infected computer network, N_B – a number of network protocol,

$Z = \{z_j\}_{j=1}^{N_Z}$ – a set of bonnets bots, N_Z – a number of botnet's bots; L – botnet's lifecycle stages; $F = \{f_j\}_{j=1}^{N_F}$ – a set of bots' functions executed during its functioning, N_F – a number of bots' functions, P - a set of ports, used by botnets.

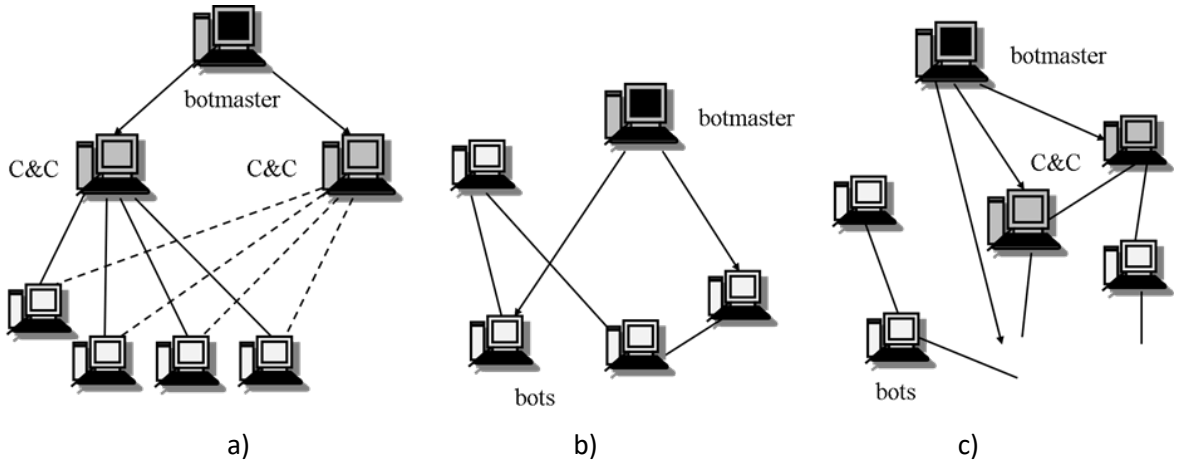

Figure 3: Botnet's architecture types: a) centralized; b) decentralized; c) hybrid

Let us consider the principles of fast-flux botnet functioning. The control elements of the "fast-flux" botnet have more advanced functionality compared to typical botnets C&C – servers, as they are hidden behind the network from a set of external proxy servers – "flux-agents". Such flux–agents are redirecting the requests to botnets' bots and their data to and from the internal control servers. In Fig.4,5 the sequences of requests and responses represented by the notation from 1 to 11 are presented.

The single fast-flux network consists of set of bots and uses the domain name *d* during the interval of DNS TTL-period. It uses the domain name *d* for the communication with the set of control elements $\{c_1, ..., c_n\}$. In addition, domain name *d* is mapped to a new cyclically changing subset of IP addresses, $d \to \{i_1, ..., i_n\}$. All mentioned IP addresses belong to the infected hosts (that is bots) that make redirection of the traffic to the control elements and are geographically distributed (in this case, no matter how), $\{c_1, ..., c_n\} := \{x \mid x \in Z \wedge x \in C\}$. Detailed scheme of single fust-flux botnet functioning is presented in Fig.4.

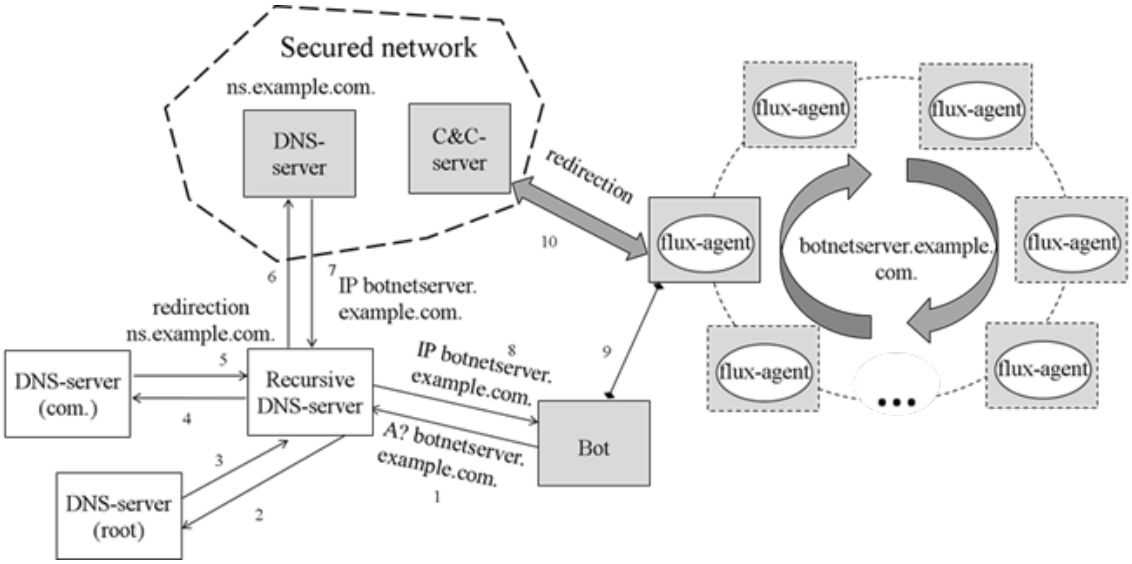

Figure 4: Single fast-flux botnet functioning scheme

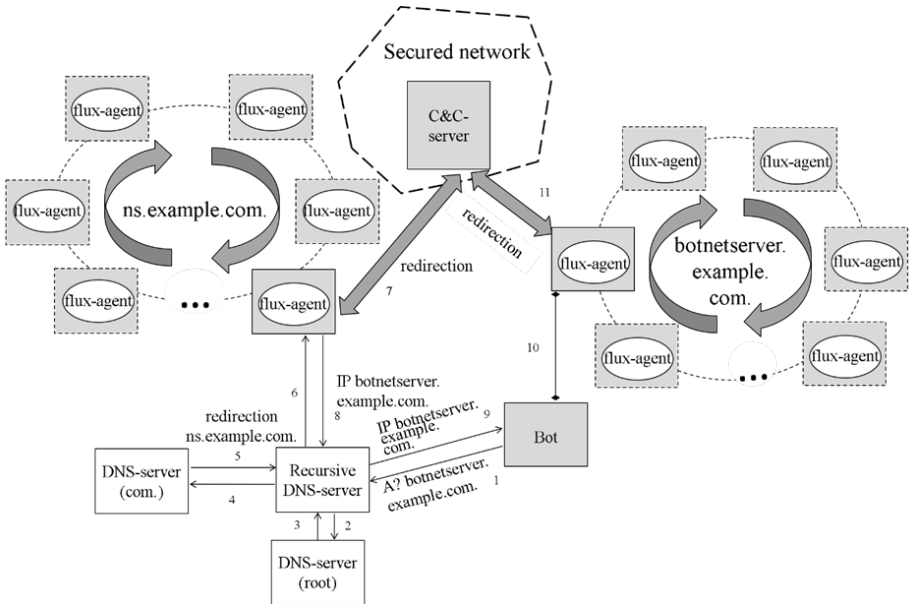

Figure 5: Double fast-flux botnet functioning scheme

For double fast-flux network (Fig.5) the addition domain names needed for the botnet construction, and that are generated by authoritative name server n are mapped to subset of cyclic changed IP addresses. In this case, we can present it as follow: $d \to \{i_1,\dots,i_n\}$, $d \to \{e_1,\dots,e_m\}$. All mentioned IP addresses additionally are geographically distributed infected computer systems, $\{n_1,\dots,n_m\} := \{x \mid x \in Z \wedge x \in N\}$. Given that, in the situation of double fast-flux botnet, amount of name servers is more than 1, $\{n_1,\dots,n_m\} \to \{e_1,\dots,e_n\}$.The use as proxies of a large number of infected compute systems located in different parts of the world, and numerous redirects make it difficult to monitor and to disable the C&C of such a botnet.

Let us present the set of computer systems which reform DNS-queries during monitoring time as $N_{\Upsilon,j,k}$ $\Upsilon = \bigcup_{j=d_1}^{d_{N_D}} \bigcup_{k=1}^{N_\tau} \Upsilon_{j,k}$, where Υ_j, $\Upsilon_{j,k}$ – subsets of MAC-addresses of the hosts, which make DNS-requests to the specific domain name during the monitoring time and within a specific TTL period respectively, $\Upsilon_{j,k} = \{\upsilon_{j,k,i}\}_{j=1}^{N_{\Upsilon,j,k}}$, where $\upsilon_{j,k,i}$ is the computer system's MAC address; N_τ – a total number of such subsets; $N_{\Upsilon,j,k}$ – the number of computer systems in the network that sent DNS queries within a certain TTL period.

Similarly, let us denote incoming DNS messages as the set $\Omega^T = \bigcup_{j=d_1}^{d_{N_D}} \bigcup_{k=1}^{N_\tau} \Omega_{j,k}$, where Ω_j – subset of captured incoming DNS messages concerning a specified domain name during the monitoring time; $\Omega_{j,k}$ – subset of the incoming messages performed by bot in the infected host, aimed to the specified domain name, captured during specified TTL period, $\Omega_{j,k} = \{\Omega_{j,k,i}\}_{j=1}^{N_{\Omega,j,k}}$, where $\Omega_{j,k,i}$ – captured DNS message from or to infected network host; $N_{\Omega,j,k}$ – amount captured messages.

Taking into account the fields of the incoming DNS message, the data from which can be used to detect DNS queries of botnets, let us describe the captured domain name DNS response as:

$$\Omega_{j,k,i} = \langle \Omega_\Upsilon, \Omega_{TS}, \Omega_{IP}, \langle \Omega_{HD}, \Omega_{ATH}, \Omega_{ADD} \rangle \rangle,$$
$$j = d_1,\dots,d_{N_D}, k = \overline{1,N_\tau}, i = \overline{1,N_{\Omega,j,k}}, \quad (2)$$

where Ω_{Υ} – MAC-address of the computer system that performed the DNS-query; Ω_{TS} – timestamp (DNS-packet capture time); Ω_{IP} – IP-address of the DNS-packet source; $\Omega_{HD}, \Omega_{ANS}, \Omega_{ATH}, \Omega_{ADD}$ – DNS-message sections: Header, Answer, Authority and Additional respectively.

$$\Omega_{HD} = \left\langle \Omega_{HD,ID}, \Omega_{HD,OPC}, \Omega_{HD,RC}, \Omega_{HD,QDC}, \Omega_{HD,ANC}, \Omega_{HD,NSC}, \Omega_{HD,ARC} \right\rangle, \quad (3)$$

where $\Omega_{HD,ID}$ – identifier that enable assigning the DNS-query and DNS-respond (ID field); $\Omega_{HD,OPC} \in \{0,...,2\}$ – query type (OPCODE field); $\Omega_{HD,RC} \in \{0,...,5\}$ – respond code (RCODE field); $\Omega_{HD,QDC}$ – a number of records in the query section (QDCOUNT field); $\Omega_{HD,ANC}$, $\Omega_{HD,NSC}$, $\Omega_{HD,ARC}$ – a number of resource queries in the respond sections, name servers and additional data (ANCOUNT, NSCOUNT, ARCOUNT fields).

Sections of respond, name servers as well as additional data has identical format and is described as:

$$\Omega_S = \left\{ \left(\Omega_{S,NM}, \Omega_{S,TP}, \Omega_{S,\tau}, \Omega_{S,RDL}, \Omega_{S,RDT} \right)_n \right\}_{n=1}^{N_{RR,S}}, \quad (4)$$

where $S \in \{"ANS","ATH","ADD"\}$, $\Omega_{S,NM}$ – the name of the domain to which the resource record belongs (NAME field); $\Omega_{S,TP}$ – the code type of the resource record (TYPE field), it defines the value and the format of the data in the RDATA field; $\Omega_{S,\tau}$ – life time of DNS records (TTL field); $\Omega_{S,RDL}$ – the length of the RDATA field (RDLENGTH field); $\Omega_{S,RDT}$ – a string that describes the resource (RDATA field); $N_{RR,S}$ – resource records' amount in DNS message section (similar value to value $\Omega_{HD,ANC}$, $\Omega_{HD,NSC}$, $\Omega_{HD,ARC}$ for corresponding section).

Let present botnet's fast-flux features:

$$\Phi = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \phi_8, \phi_9, \phi_{10}, \phi_{11}, \phi_{12}\}, \quad (5)$$

where ϕ_1, ϕ_2, ϕ_3 – TTL-period, mode, median, average value respectively, ϕ_4 – a number of A-records of DNS-messages; ϕ_5 – average distance between IP concerning incoming DNS-message's A-records; ϕ_6 – amount of unique IP concerning A-records; ϕ_7 – average distance between unique IP of the incoming DNS-messages, ϕ_8 – amount of various autonomous system numbers; ϕ_9 – average distance between IP for NS-records; ϕ_{10} – amount of various autonomous system numbers for name servers; ϕ_{11} – amount of NS-records; ϕ_{12} – DNS retry timeout,

The presence of the botnet that uses the fast-flux evasion technique based on the DNS can be detected using a rule:

$$\begin{aligned} &if \left(\phi_1 \in [0.900] and\ \phi_2 \in [0.900] and\ \phi_3 \in [0.900]\right) and \\ &and \left(\left(\phi_4 \in [5.\infty] and\ \phi_5 \in [65535.\infty]\right)\right) or \left(\phi_6 \in [8.\infty]\right) and \\ &and\ \phi_7 \in (65535,\infty)\big) or\ \phi_8 > 2\big) and \left(\phi_9 \in (65535,\infty) or\right. \\ &or\ \phi_{10} > 2\ and\ \phi_{11} > 3\ and\ \phi_{12} \in [0.900]\big)\big) \Rightarrow fast\_flux. \end{aligned}, \quad (6)$$

3.3. Fast-Flux Botnet Detection Process

Let present botnet's fast-flux detection procedure as:

$$M_D = \left\langle \Omega^T, f_1^T, C_{GA}^T, C_{ET}^T, f_2^T, T \right\rangle, \quad (7)$$

where Ω^T – set of gathered DNS responds to the set of hosts in the network; f_1^T – comparison procedure of domain names in "white/black" lists; C_{GA}^T – algorithms for botnet detection that use the group activity in DNS-traffic [28,29]; C_{ET}^T – algorithms for botnet detection, that use the fast-flux evasion technique [30]; f_2^T – a function for the infected computer systems localization and the bots' blocking; $T = \{t_m\}_{m=0}^{N_T}$ – monitoring time, where N_T – a number of the monitoring iterations.

The scheme of the fast-flux botnet detection is presented in Fig.6.

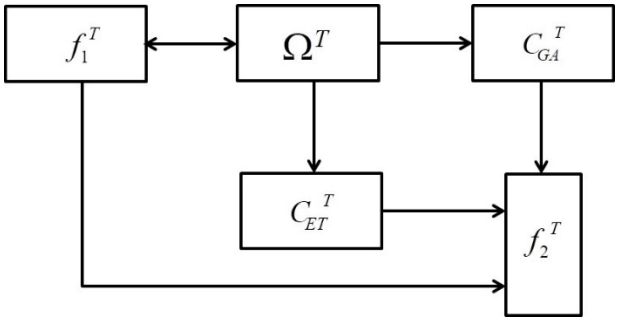

Figure 6: The scheme of the fast-flux botnet detection

Fast-flux botnet detection process as the time diagram is given in Fig.7, where t_o – the start of the monitoring (the incoming DNS-traffic gathering), $\{t_1,...,t_n\}$ – the monitoring iterations time.

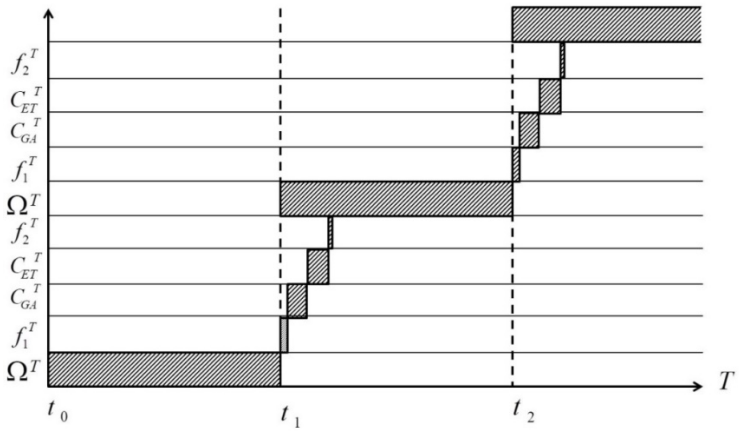

Figure 7: The time diagram of the fast-flux botnet detection

4. Experiments

4.1. Evaluation Setting

To perform the model's validation with the employment of the botnet datasets [31-35], number of experiments were conducted.

The main aim of the dataset's usage was to combine all possible botnets functioning aspects (different botnets' features, architectural properties, their functional intent). To do this the set of botnets' traffic dump were analyzed and the set of botnets' behavior models at different botnets' lifecycle stages were built. The training data contained 132415 samples of fast-flux DNS traffic, and the test data contained 32415 samples. Among a great number of botnets' samples such as TrickBot, Lokibot, AZORult, NanoCore, NetWire, Gozi, RemocsRAT, ArkeiStealer, NjRAT and other [36] were used for the experiments. Also test data contained 16804 samples of benign traffic.

To perform the classification procedure the set of classifiers were employed: Support Vector Machine [24], k-nearest neighbors [38], fuzzy c-means clustering [28, 39], Artificial Immune System [37, 40]. As the classification core the framework BotGRABBER [24] was used.

4.2. Experiments Results

To test the effectiveness of the proposed approach, all the test samples were divided into the classes FL1-FL6, according to their nature:

1. FL1 – a set of centralized single-flux botnets bots;
2. FL2 – a set of centralized double-flux botnets bots;
3. FL3 – a set of decentralized single-flux botnets bots;
4. FL4 – a set of decentralized double-flux botnets bots;
5. FL5 – a set of hybrid single-flux botnets bots;
6. FL6 – a set of hybrid double-flux botnets bots.

Test result of experiments are presented in figures 8-10.

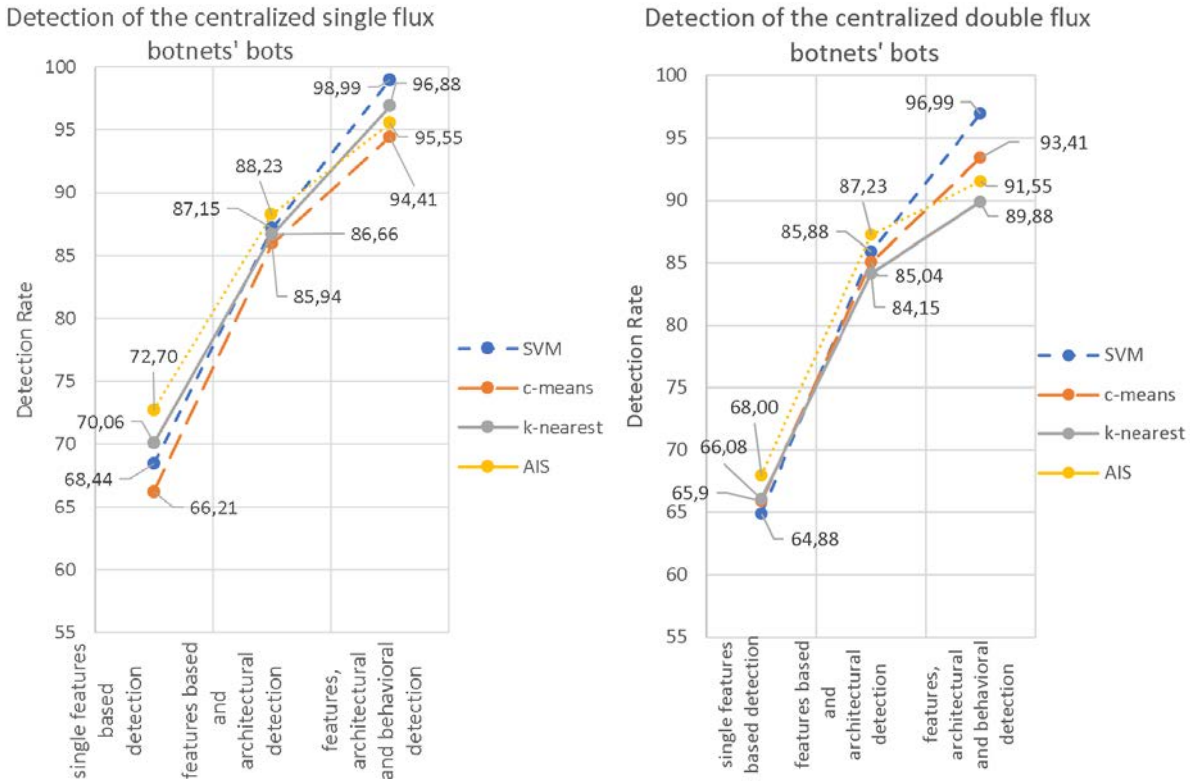

Figure 8: Detection results for a) the centralized single-flux botnets' bots; b) the centralized double-flux botnets' bots

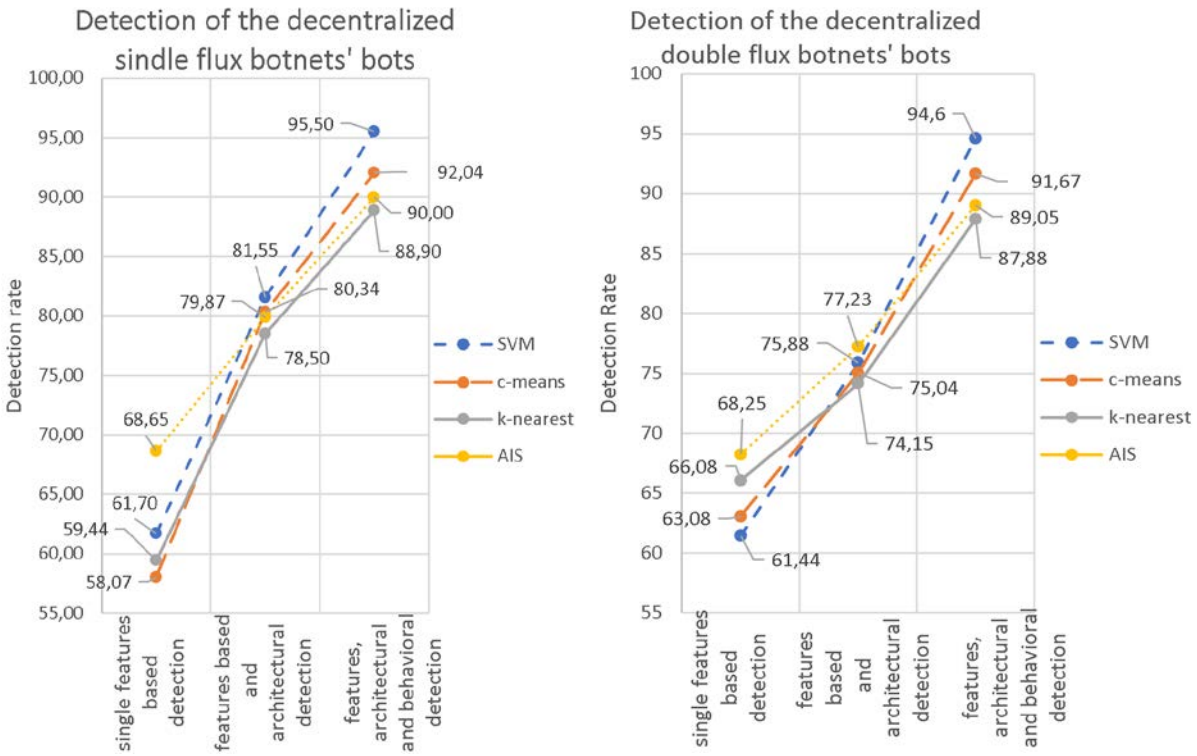

Figure 2: Detection results for a) the decentralized single-flux botnets' bots; b) the decentralized double-flux botnets' bots

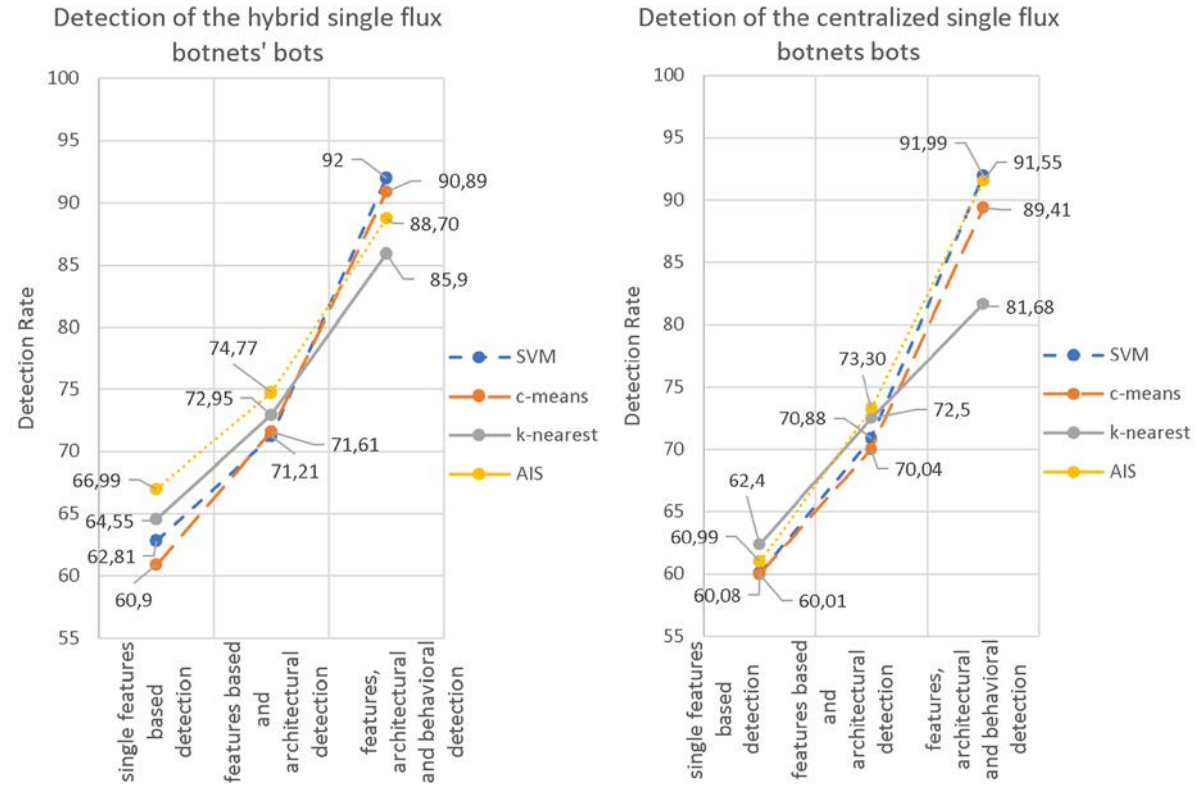

Figure 3: Detection results for a) the hybrid single flux-botnets' bots; b) the hybrid double-flux botnets' bots

Test result of experiments illustrated that involvement the only analysis of features demonstrated low detection efficiency. Involvement of the analysis of the architectural botnets' aspect into the detection process increased the efficiency up to 88%, while addition of the botnets' bots behaviors analysis allowed increasing of the detection efficiency up to 99%. The false positives were at the rate from 3 to 6%.

Experiments also showed that the botnets with centralized single-flux architecture are easy to detect, while the detection results concerning hybrid are rather lower.

Another aspect of the experiments results is the involvement of different classifiers. The difference between classification results among used classifiers is not high but overall results is significant.

5. Discussion

Proposed technique can be improved by adding new malicious samples of the fast flux botnets' attacks. To increase the detection accuracy different machine learning algorithms may be employed. In addition, the technique can be improved by addition the new features and behavioral aspects, relevant to the botnets' attacks and providing correspond security scenarios.

6. Conclusion

The paper presents DNS-based fast-flux botnet detection approach. The detection process is based on botnets' functioning models. Proposed models describe the bot-nets' functioning and take into consideration features analysis, architectural particularities, as well as botnet's behavior such as group activity in the network and hosts.

Proposed botnets' models take into account the use of botnets DNS at theirs life cycle stages, and the use of botnet fast-flux technology to avoid detection.

Another aspect of the approach was to take into consideration different ways of communication of bots with command-and-control centers of botnets and to identify botnets' bot by analysis of theirs architecture (centralized, distributed and hybrid). Combination of mentioned aspect enabled to increase the detection efficiency.

As the mean of conclusion making different classifiers were involved. Thus, experiments have shown the ability to detect botnets involving the developed technique up to 99%, while the false positives level was about 3-6%.

7. References

[1] R. Leizerovych, G. Kondratenko, I. Sidenko and Y. Kondratenko, "IoT-complex for Monitoring and Analysis of Motor Highway Condition Using Artificial Neural Networks," 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, pp. 207-212 (2020), doi: 10.1109/DESSERT50317.2020.9125004.

[2] Eset. Spyware. Available online: https://help.eset.com/glossary/en-US/spyware.html (accessed on January 20, 2021).

[3] Avast. Spyware: Detection, Prevention, and Removal. Available online: https://www.avast.com/c-spyware (accessed on January 20, 2021).

[4] Securelist. New trends in the world of IoT threats. Available online: https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/ (accessed on January 20, 2021).

[5] Sokol P., Zuzčák M., Sochor T. Definition of attack in the context of low-level interaction server honeypots. Lecture Notes in Electrical Engineering 330, pp. 499–504 (2015).

[6] S. Scanzio, L. Wisniewski, i P. Gaj, „Heterogeneous and dependable networks in industry – A survey", Computers in Industry, t. 125, s. 103388, luty 2021, doi: 10.1016/j.compind.2020.103388.

[7] O. Drozd, K. Zashcholkin, O. Martynyuk, O. Ivanova, J. Drozd. Development of Checkability in FPGA Components of Safety-Related Systems. CEUR Workshop Proceedings, vol. 2762, pp. 30-42 (2020). Online http://ceur-ws.org/Vol-2762/paper1.pdf.

[8] Check Point Research. The 2020 Cyber Security Report. Available online: https://research.checkpoint.com/2020/the-2020-cyber-security-report/ (accessed on January 20, 2021).
[9] Melnyk, A., Melnyk, V. Remote Synthesis of Computer Devices for FPGA-Based IoT Nodes. 2020 10th International Conference on Advanced Computer Information Technologies, ACIT 2020 – Proceedings 9208882, pp. 254-259.
[10] A. Cabri, G. Suchacka, S. Rovetta and F. Masulli. Online Web Bot Detection Using a Sequential Classification Approach. 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, United Kingdom, pp. 1536-1540 (2018), doi: 10.1109/HPCC/SmartCity/DSS.2018.00252.
[11] Mousavi, S. H., Khansari, M., & Rahmani, R. A fully scalable big data framework for botnet detection based on network traffic analysis. Information Sciences, 512, pp. 629-640 (2020).
[12] Derakhshan, F., & Ashrafnejad, M.The risk of botnets in cyber physical systems. In Security of Cyber-Physical Systems, pp. 81-106. Springer, Cham (2020).
[13] Al-Nawasrah, A., Almomani, A. A., Atawneh, S., & Alauthman, M. A Survey of Fast Flux Botnet Detection With Fast Flux Cloud Computing. International Journal of Cloud Applications and Computing (IJCAC), 10(3), pp. 17-53 (2020).
[14] Surjanto, W., & Lim, C. Finding Fast Flux Traffic in DNS Haystack. In International Conference on Critical Information Infrastructures Security, pp. 69-82. Springer, Cham (2020).
[15] Li, Wanting, Jian Jin, and Jong- Hyouk Lee. Analysis of Botnet Domain Names for IoT Cybersecurity. IEEE Access 7: 94658-94665 (2019).
[16] Alieyan, K., Almomani, A., Anbar, M., Alauthman, M., Abdullah, R., Gupta, B. B. DNS rule-based schema to botnet detection. Enterprise Information Systems, pp. 1-20 (2019).
[17] Singh, M., Singh, M., & Kaur, S. Issues and challenges in DNS based botnet detection: A survey. Computers & Security, 86, pp. 28-52 (2019).
[18] Almomani, A. Fast-flux hunter: a system for filtering online fastflux botnet. Neural Computing and Applications, 29(7), pp. 483-493 (2018).
[19] Al-Nawasrah, A., Al-Momani, A., Meziane, F., & Alauthman, M. Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm: In 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 7-11). IEEE (2018).
[20] Zang, X. D., Gong, J., Mo, S. H., Jakalan, A., & Ding, D. L. Identifying fast-flux botnet with AGD names at the upper DNS hierarchy. IEEE Access, 6, 69713-69727 (2018).
[21] Lombardo, P., Saeli, S., Bisio, F., Bernardi, D., & Massa, D. Fast flux service network detection via data mining on passive DNS traffic. In International Conference on Information Security, pp. 463-480. Springer, Cham (2018).
[22] Alieyan, K., Anbar, M., Almomani, A., Abdullah, R., & Alauthman, M. Botnets Detecting Attack Based on DNS Features. In 2018 International Arab Conference on Information Technology (ACIT) (pp. 1-4). IEEE (2018).
[23] Lysenko, S., Savenko, O., Bobrovnikova, K., Kryshchuk, A.: Self-adaptive System for the Corporate Area Network Resilience in the Presence of Botnet Cyberattacks. In: International Conference on Computer Networks, pp. 385-401. Springer, Cham (2018).
[24] Lysenko S., Bobrovnikova K., Savenko O., Kryshchuk A. BotGRABBER: SVM-Based Self-Adaptive System for the Network Resilience Against the Botnets' Cyberattacks. In: Gaj P., Sawicki M., Kwiecień A. (eds) Computer Networks. CN 2019. Communications in Computer and Information Science, vol 1039, p. 127-143. Springer, Cham (2019).
[25] Sochor, Tomas. Detection Efficiency Improvement in Multi–component Anti-spam Systems. In: International Conference on Computer Networks. Springer, Cham, p. 91-100 (2020).
[26] Zuzčák, Matej; Sochor, Tomáš; Zenka, Milan. Intrusion Detection System for Home Windows based Computers. KSII Transactions on Internet & Information Systems, 13.9 (2019).
[27] O. Savenko, S. Lysenko, A. Nicheporuk and B. Savenko, "Approach for the unknown metamorphic virus detection," 2017 9th IEEE International Conference on Intelligent Data

Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017, pp. 71-76, doi: 10.1109/IDAACS.2017.8095052..
[28] Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A., Bobrovnikova, K. A Technique for the Botnet Detection Based on DNS-Traffic Analysis. In: Gaj P., Kwiecień A., Stera P. (eds) Computer Networks. CN 2015. Communications in Computer and Information Science, vol 522, p. 127-138. Springer, Cham (2015).
[29] Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A., Bobrovnikova, K.: Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing. In: International Conference on Computer Networks: Springer International Publishing, pp. 83-95. Springer, Cham (2016).
[30] Lysenko, S., Pomorova, O., Savenko, O., Kryshchuk, A. and Bobrovnikova, K. DNS-based anti-evasion technique for botnets detection. 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, 2015, pp. 453-458 (2015), doi: 10.1109/IDAACS.2015.7340777.
[31] Canadian Institute for Cybersecurity. Botnet dataset. Available online: https://www.unb.ca/cic/datasets/botnet.html (accessed on September 10, 2020).
[32] University of Victoria. ISOT Research Lab. Botnet and Ransomware Detection Datasets. Available online: https://www.uvic.ca/ engineering/ece/isot/datasets/botnet-ransomware/index.php (accessed on September 10, 2020).
[33] The BoT-IoT Dataset. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php (accessed on September 10, 2020).
[34] CAIDA. Center for Applied Internet Data Analysis. https://www.caida.org/home/ Available online: (accessed on September 10, 2020).
[35] Koroniotis, N., Moustafa, N., Sitnikova, E., \& Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems, 100, pp. 779-796 (2019).
[36] The Spamhaus Projec. Botnet Threat Report 2019. Available online: https://www.spamhaus.com/custom-content/uploads/2020/04/2019-Botnet-Threat-Report-2019-LR.pdf (accessed on January 20, 2021).
[37] Dasgupta, Dipankar (ed.). Artificial immune systems and their applications. Springer Science & Business Media (2012).
[38] Prasath, V. B., Alfeilat, H. A. A., Hassanat, A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., & Salman, H. S. E. Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier-A Review. arXiv preprint arXiv:1708.04321 (2017).
[39] Sergii Lysenko, Oleg Savenko, Kira Bobrovnikova. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. CEUR-WS, ISSN: 1613–0073 (Scopus). 2018. Vol. 2104. Pp. 688-695.
[40] Lysenko, S., Bobrovnikova, K., Shchuka, R., & Savenko, O. (2020, May). A Cyberattacks Detection Technique Based on Evolutionary Algorithms. In 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies. pp. 127-132.