A System for Detecting Anomalies and Identifying Smart Home Devices Using Collective Communication

Anastasiia Nicheporuk[a], Andrii Nicheporuk[a], Anatoliy Sachenko[b], Oleg Sachenko[b] and Andrii Kazantsev[a]

[a] *Khmelnitsky National University, 11 Instytutska Str., 29016, Ukraine*
[b] *West Ukrainian National University, 11 Lvivska Str., 46009, Ukraine*

Abstract

The fourth industrial revolution put on new rails processes of automation in industry, healthcare, home and other areas of human life through the mass integration of the concept of the Internet of Things into these areas. However, this concept leaves a number of potential "bottlenecks" in the security of such systems for attackers. Third-party access to data collected by smart devices in, for example, a smart home can lead to a variety of emergencies, the degree of danger of which will depend solely on the will of the owner of the intercepted data. In this paper we proposes a system for detecting anomalies and identifying smart home devices based on the collective communication of smart homes. The concept of the system is based on the benefits of combining smart homes into a social network in terms of improving the security of both a single smart home and the entire social network of combined smart homes. Detection of anomalies and identification of devices in each of the smart homes is based on monitoring network traffic and forming profiles of smart devices that are present in the network. Based on this, a whitelist of allowed profiles of devices operation in the cluster is formed. To verify the presence of a profile in the whitelist the Random Forest algorithm was used. A key feature of the system is the communication of smart home clusters with each other to exchange information about the available smart device profiles in the whitelists of each cluster. To evaluate the effectiveness of the proposed system, a number of experimental studies were conducted. The results of the experiments showed the overall accuracy of the system at the level of 97.21% with an average level of type I errors of 5.94%.

Keywords

Collective Communication, Smart home, Smart device, Traffic, Profile, White list

1. Introduction

For a considerable period of time, humanity get the benefit from the use of smart devices connected to a network to improve and automate life. Smart homes, healthcare systems, automation systems, and Industrial Internet of Things (IIoT) are prime examples of systems based on a network of connected smart devices under the control of microcontrollers or FPGA [1-5]. The next step in the evolution of such systems is their integration into social networks. For example, the integration of smart homes into a social network forms a higher hierarchical level of their interaction, which produces new opportunities and benefits, in particular in terms of management, storage and processing of information, improving end-user service, prevention and collective response to emergencies and events, in terms of security, etc. Unfortunately, in terms of the security of such

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine
EMAIL: eldess06@gmail.com (A. Nicheporuk); andrey.nicheporuk@gmail.com (A. Nicheporuk); as@wunu.edu.ua (A. Sachenko); os@wunu.edu.ua(O. Sachenko); andreykaololo@gmail.com (A. Kazantsev)
ORCID: 0000-0001-5366-5792 (A. Nicheporuk); 0000-0002-7230-9475 (A. Nicheporuk); 0000-0002-0907-3682 (A Sachenko); 0000-0001-9337-8341 (O. Sachenko)

networks, the integration of smart homes into social networks does not remove their vulnerabilities, which were inherent in their constituent components at a lower level.

Our research focuses on benefiting from the integration of smart homes into a social network in terms of improving the security of both a single smart home and the entire social network of combined smart homes. We represent each smart home as a cluster, which is an automated system with a set of smart devices that are connected to each other and the gateway, communication between which is carried out by involving a stack of TCP / IP protocols. The mechanism of cluster interaction is one-to-many, i.e. from one cluster, through a router with Internet access, it is possible to receive / transmit information to / from all other clusters. Thus, the task of the study is to decide on the absence / presence of abnormal behavior in the communication environment of a smart home by monitoring its network traffic and identifying network data flows from specific devices, as well as, if necessary, attracting information about abnormal activity from other clusters which are part of the social network of collective communication.

2. Vulnerabilities and attacks on IoT networks

From year to year, the attackers try to compromise and stole the private information by hacking the local and corporate networks. From the point of view of IoT networks, the specifics of their work reveal even more vulnerabilities and existing bottlenecks for attacks in contrast with "conventional" network [6]. The most important reasons why criminals choose to attack IoT devices, in particular smart homes, are their constant availability on the Internet, limited computing capabilities, which makes it impossible to install security systems directly on the devices themselves, vulnerabilities related to authorization / authentication devices on the network, heterogeneity of the devices and their communication environment, vulnerabilities in web interfaces, lack of proper attention from end users ("put and forget", which often manifests itself in leaving standard logins and passwords, lack of checking for updates, etc.) [7-10]. The combination of these factors leads to considerable interest among criminals who are trying to implement more and more cyber-attacks.

The main types of attacks on IoT networks are DoS / DDoS attacks and Man in the middle (MITM) attacks. Their common goal is to capture control of the device and use it for their own purposes. These goals can be, for example, the participation of the device in the botnet (Mirai botnet [11]), reading, interception, distortion of information by compromising the communication channel (e.g. ARP Spoofing attack), disconnecting the device from the network, thereby disrupting the established process operation of the whole system (for example, disabling surveillance cameras, motion sensors, etc.) [12, 13]. In addition, the simplicity and flexibility of introducing new smart devices, applications and services into the Smart House system makes them look like building blocks [14-16], lack of understanding of internal operation of which can lead to unknown vulnerabilities and their spread on a large scale.

3. Related works

Today, considerable attention is paid to the problem of detecting cyber-attacks on the IoT network. Existing intrusion detection systems in the IoT network can be divided into several main groups: intrusion detection systems based on signatures [17], intrusion detection systems based on rules and anomaly detection systems [18]. One of the most promising areas is the detection of anomalies in network traffic. The anomaly detection technique is to learn legitimate behavior from the normal network traffic and identify the variations from it. Since anomaly detection just inspects deviations from the benign traffic rather than the attack signatures, it has the capability to identify the zero-day attacks as well [19-21]. Known approaches of abnormal activity detection in network traffic primarily differ in the choice of features or parameters that determine the difference between normal and harmful profile, the way they are presented and methods of processing. Let's take a closer look at some of them.

In [22] authors propose the IOT-KEEPER, an edge system capable of performing online traffic classification at network gateways. The system represents the traffic with features that are agnostic with respect to the IoT communication technology, but only depends on TCP/IP features which can be observed by the edge. IOT-KEEPER uses fuzzy C-Means clustering in order to identify different types

of IoT device activities based on their network footprints. The framework then uses these properties and characterizations to identify anomalies in device activities by analyzing its network traffic.

Authors in [23] presents the possibility of using the Hurst coefficient to determine the level of self-similarity of the traffic, which affects the ability to determine the typical operating states as well as the detection of certain anomalies such as an attack, refusal of access, overload and post-failure state. Additionally, authors present results of an analysis of traffic in the communication network using a statistical coefficient of similarity and multifracture spectrum. The presented results of the measurements and research confirmed that the analyzed traffic was self-similar and amounted to 0.5–1.

Another anomaly based approach is presented in [24]. The authors utilize a deep packet analysis approach which employs a bit-pattern technique. The network payloads are treated as a sequence of bytes called bit-pattern, and the feature selection operates as an overlapping tuple of bytes called n-grams. When the corresponding bits matches all positions, a match between the bit-pattern and n-grams occurs. The system is evaluated by deploying four attacks and demonstrates a very low false-positive rate.

In [25] the authors proposed a two-level anomalous activity detection model for intrusion detection system in IoT networks. The first layer of the proposed model categorizes the network flow as normal flow or abnormal flow using decision tree classifier. If the detected flow is anomalous, then the first level transfers the flow to the second layer. The second layer classifies the type of this attack using random forest algorithm. Proposed solution can characterize the network traffic as Normal, DDoS-HTTP, DDoS-TCP, DDoS-UDP, DoS-HTTP, DoS-TCP, DoS-TCP, OS-Fingerprint, Service-Scan, Keylogging, and Data-Exfiltration.

A method to detect anomalous operations by learning user behaviors in smart home is presented in [26]. Proposed method uses Hidden Markov Models to learn the normal activities of a user. This method uses the information obtained from sensors of the home IoT devices as the observations. By using the observations, this method learns the parameters of Hidden Markov Models. Next, this method detects the anomalous operations if an operation whose probability is low occurs. Authors demonstrated the accuracy of this method by using the dataset collected at the smart home environment deployed by them.

Another similar method that learns user behaviors and focuses on the scenario of interaction of multiple users with smart home devices is presented in [27]. This method models user behavior as sequences of user events including operation of home IoT devices and other monitored activities. Considering users behave depending on the condition of the home such as time and temperature, this method learns event sequences for each condition. To mitigate the impact of events of other users in the home included in the monitored sequence, presented method generates multiple event sequences by removing some events and learning the frequently observed sequences.

The presented solutions have shown a fairly high level of efficiency in detecting abnormal behavior in IoT networks, but they are primarily focused on protecting of a single ecosystem of a smart home, not taking into account the possible security gains in the case of combination of clusters of smart homes into a single social network.

4. Profiling of smart home devices

Despite the identity of the physical environment and the protocol for transmitting information in TCP / IP networks, the characteristics of network traffic for smart devices and conventional non-intelligent nodes will be different. This situation is explained by the nature and main purpose of smart devices. The main purpose of smart devices is to periodically monitor environmental processes and exchange information with other smart devices or end users (the end users are other smart devices or various user devices – computers, tablets, smartphones, etc.). Given the specifics of smart devices, the following characteristics of network traffic can be distinguished: activity period, sleep period, packet size and amount of information transmitted within the session, frequency and number of DNS queries [28-31].

The period of activity reflects the phase of the life cycle of a smart device, which is manifested in its active network interaction with other devices in the network. This activity is the transmission of information about the parameters of the physical environment or synchronization signals to maintain communication with other participants in the M2M interaction. Accordingly, this period will be characterized by a surge in network activity produced by a smart device. A review of previous studies

has shown [28] that for a significant number of smart devices, including TP-Link Smart Plug Switch, the duration of this period is not more than 5 seconds.

The activity of the devices is changed by sleep periods during which the exchange of packets in the network with the participation of a smart device and other devices in the network is absent. For most smart devices, this period is no more than 20 seconds.

The size of packets and the amount of information transmitted within a session involving a smart device can also be a key aspect in profiling (behavioral differentiation) devices of smart homes. Typically, the packet size of data transmitted by smart devices is small. At the same time, the amount of information transmitted within one session is not more than 1 KB.

Another feature that allows you to differentiate smart device profiles is the number and frequency of DNS queries. Due to the highly specialized nature of the operation of smart devices, the frequency and number of DNS queries is not high. Smart devices often use the domain names of smart device manufacturers, for example, the Amazon Echo produce DNS queries to softwareupdates.amazon.com, device-metrics-su, amazon.com, example.org, pindorama.amazon.com, and pool.ntp.org [20]. While the LiFX lightbulb communicates with only two domains v2.broker.lifx.co and pool.ntp.org.

Thus, understanding the features of the interaction of smart devices allows us to identify a set of features (or attributes) that can be used to describe the behavior and characteristics of a smart device in a smart home.

5. A system for detecting anomalies and identifying smart home devices using collective communication

The implementation of the system for detecting anomalies and identifying devices in smart homes is based on monitoring of network traffic and construction of profiles of smart devices that are present in the network. Based on this, a whitelist of allowed profiles of devices operation in the cluster is formed. A key feature of the system is the communication of clusters with each other to exchange information about the available profiles of smart devices. The generalized functional scheme of the system for detecting anomalies and identifying devices in smart homes is given in fig. 1. Let's take a closer look at the components of the proposed system.

The system of detection of anomalies and identification of devices is placed in the internal network of a smart home and consists of the following modules: network traffic monitoring, detection of anomalies in network behavior, identification and decision making module, PDML conversion module, features extraction module, classification module. Information about available devices in smart home is stored in a white list of smart device profiles. It is assumed that smart homes are connected to a social network.

The identification and decision making module organizes the operation of the system in three modes: monitoring of network traffic and detection of anomalies; search for device profiles in the cluster; search for a profile in other clusters (Fig. 1). Let $D=\{d_1,d_2,...,d_n\}$ be the set of smart devices, whitelisted and connected in cluster *C*, where $C=\{c_1,c_2,...,c_m\}$, m is the number of clusters. Assume that in each cluster, all smart devices are whitelisted, i.e. the whitelist was generated immediately after all devices were connected. In addition, it is assumed that the smart devices were operated in normal mode, without performing firmware change, reconfiguration or other similar operations. Also, let each cluster have its own set of smart devices, and accordingly, its own whitelist, which may differ from those existing on other clusters.

In the first mode, abnormal behavior in network traffic is tracked. If abnormal behavior is detected, the transition of system to the second mode is carried out, in which the classification of network traffic and identification of smart devices is performed. Let a structured set of traffic data is setted, then the result of the system for a given stream of IP packets (sessions) is it's comparison with the set *D* in the cluster c_j and determine d_i for which behavior of a given stream of packets is closest. If as a result of classification it was possible to match a suspicious profile to one of profiles d_i of smart devices in smart home, then the system returns to the first (regular) mode of operation.

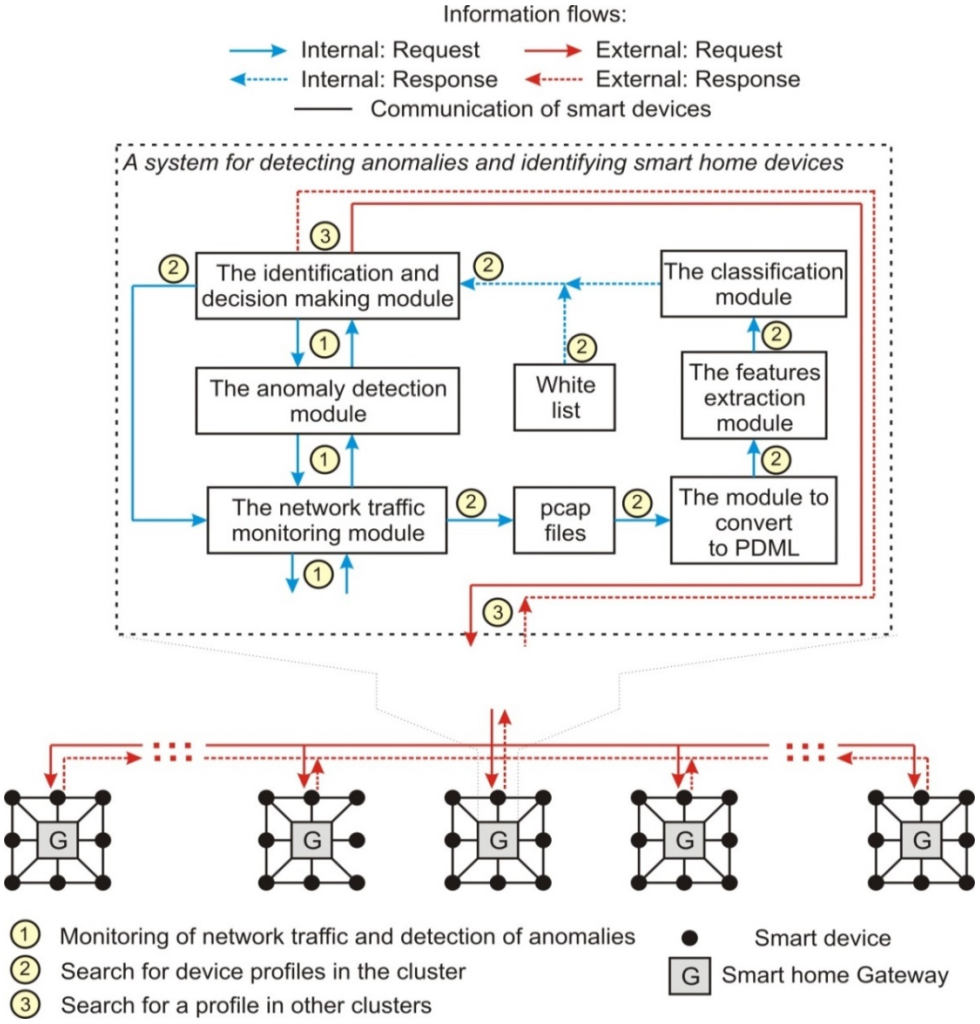

Figure 1: The generalized functional scheme of the system for detecting anomalies and identifying devices in smart homes

Otherwise, if the specified sequence of packets in the cluster c_j is not matches to one of the profiles d_i (the threshold value is not exceeded), the sequence of packets is denoted as "unknown sequence of packets" and the system have been transitioning into the third mode. This mode involves requesting to other clusters $c_k, 1 < k \le m-1$ in order to check the sequence of packets (theirs profile) received in the cluster c_j with whitelists in each cluster in the social network.

After performing of classification and identification on each of the clusters c_k, the all results are sent to the cluster c_j, where the module of identification and decision making makes the final conclusion. To form a conclusion, the module of identification and decision making in the cluster c_j from the results of the responses of all clusters forms a list in which each element is either a type of device in the found cluster $d_i^{c_k}$ or "unknown sequence of packets". The result of the system $\phi(c_j)$ in the cluster c_j for the generated list is calculated as:

$$\phi(c_j) = \frac{\sum_{k=1}^{k \le m-1} h_{c_k} \cdot w_{c_k}}{\sum_{k=1}^{k \le m-1} u_{c_k} \cdot w_{c_k}} \qquad (1)$$

where m – the number of clusters;

h_{c_k} – the result of a cluster c_k in which the sequence of packets from the cluster c_j is defined as the type of smart device contained in the white list of the cluster c_k, $h_{c_k} = \{ x \mid x \in 0 \vee 1 \}$;

u_{c_k} – the result of a c_k cluster in which the sequence of packets from the cluster c_j is defined as "unknown sequence of packets", i.e. the behavior profile is missing in the white list of the cluster c_k, $u_{c_k} = \{ x \mid x \in 0 \vee 1 \}$;

w_{c_k} – the weighting coefficient of the importance of the result.

The weighting coefficient of the importance of the result for the cluster c_k is calculated as the ratio of the number of smart devices in the cluster c_k to the total number of all smart devices in all clusters [29]:

$$w_{c_k} = \frac{p_{c_k}}{\sum\limits_{i \le m} p_{c_i}}, \tag{2}$$

where p_{c_k} is the number of smart devices in the cluster whitelist c_k; p_{c_i} – the number of smart devices in the cluster whitelist c_i;

If the result of the system $\phi(c_j)$ is greater than one, then the sequence of packets that form a suspicious profile could be attributed to one of the profiles of smart devices d_i in one of the clusters in the social network:

$$\phi(c_j) = \begin{cases} d_i^{c_k} & if\ \phi(c_j) \ge 1 \\ \text{"unknown"} & if\ \phi(c_j) < 1 \end{cases}, where\ c_j \in C, c_k \in C. \tag{3}$$

The network traffic monitoring module is used to scan traffic generated by smart devices and receive a sequence of TCP packets. The data is stored as pcap files, each of which is a set of TCP sessions. The beginning and end of the session were determined by the SYN and FIN flags, respectively. Within each session, packets are grouped based on four header fields:

$$\langle src\_ip, src\_port, dst\_ip, dst\_port \rangle, \tag{4}$$

where src_ip – source IP address, src_port – source port number, dst_ip – destination IP address, dst_port – destination port number.

The network traffic monitoring module is involved both for tracking anomalous activity (as part of the anomaly detection module) in real time and for obtaining "raw data" of network traffic, which will be further processed to prepare for the process of classifying network traffic.

The anomaly detection module is used to track critical network characteristics in real time. The operation of this module can be represented as a trigger that generates an alarm in case of suspicious characteristics. These characteristics are the amount of traffic, bandwidth usage, increasing the number of connections to / from one TCP port, increasing the number of connections to one / from one IP address. The value of the above characteristics is obtained on the basis of a statistical assessment of the network over time from the moment of initial setup and connection of all smart devices to the network.

In order to check the presence of a profile in the white list, which describes the sequence of packets of network traffic specific to each of the smart devices d_i, the *classification module* is used, which is based on the Random Forest algorithm [33]. In order to obtain features for the classification (i.e. the profile of the behavior of the flow of network packets), the entire fixed period of traffic monitoring *T* was divided into time intervals t_i, each of which consisted of a set of sessions *s* (Fig. 2). If session s started within a time interval t_i but ended outside that time interval, such a session was related to the interval t_i. If session *s* ended before the end of the time interval t_i, the time before the next session referred to the time interval t_i.

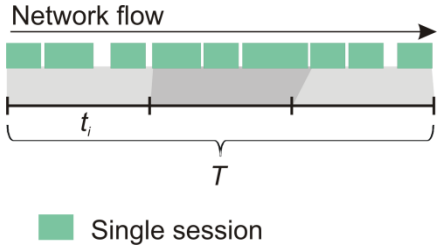

Figure 2: Separating the monitoring period into time intervals

Then, within each interval t_i, a set of features is obtained $F = \langle f_1, f_2, ..., f_{10} \rangle$ that describe the behavior of the packet flow given the specifics of the operation of smart devices (Table 1). It should be noted that since payload is encrypted, none of the features take this field into account.

Table 1

Features used to describe the behavior of the packet flow

№	Features
1	Period of activity
2	Sleep period
3	Number of DNS queries
4	Frequency of DNS queries
5	Average package size
6	Maximum package size
7	Ratio of transmitted-bytes to received-bytes
8	Sum of amount received-bytes
9	Sum of amount of transmitted-bytes
10	Time between the first and last packet
11	NTP interval

To do this, all collected network traffic is converted to Packet Description Markup Language (PDML) using *the module to convert to PDML* [34]. PDML represents packet header fields in XML format, which allows to access all the attributes of packets that are used as component part of features.

As a result of the classifier for each time interval t_i, which is represented by the vector *F*, a probability vector is obtained, each element of which determines the belonging of the packet flow to a smart device d_i in the cluster $c_j, r^{c_j}(t_i) = \left\langle r_{d_1}^{c_j}, r_{d_2}^{c_j}, ..., r_{d_n}^{c_j} \right\rangle$, so that $\sum_{i=1,j=1}^{n,m} r_{d_i}^{c_j} = 1$ where *m* is the number of clusters in the social network, *n* is the number smart devices in the c_j cluster.

A packet sequence is considered to belong to one of the smart devices d_i if $r_{d_i}^{c_j} \geq \delta$, where δ is the probability threshold which determines the belonging to the class of smart device d_i in the cluster c_j.

The final choice of a smart device d_i in the cluster c_j (or the designation of the collected profile as an "unknown sequence of packets") for the entire monitoring period *T* is defined as the modal value from all intervals t_i. Thus, *the white list of profiles* contained in each cluster is in fact a trained classifier model and contains a marked set of features for a given monitoring period *T* and given time intervals t_i.

Thus, based on the use of collective communication to exchange information about the search for a suspicious profile in the whitelists of other network clusters, a decision of the absence / presence of abnormal behavior in the communication environment of a smart home is made. Knowledge of this

information allows the user or network administrator to perform preventive actions to block network traffic or disconnect a device from the network that produces abnormal activity.

6. Experiments and evaluation

To conduct an experiment to determine the effectiveness of the proposed system for detecting anomalies and identifying devices of smart homes using collective communication, a social network with two clusters was deployed (Fig. 3). In addition to the router and RPi in each cluster, seven smart devices were involved, four of which were located in the first cluster, the remaining three in the second (Table 2). To collect network traffic generated by smart devices, the Tshark utility [35] was used, which was installed on RPi running Raspberry Pi OS.

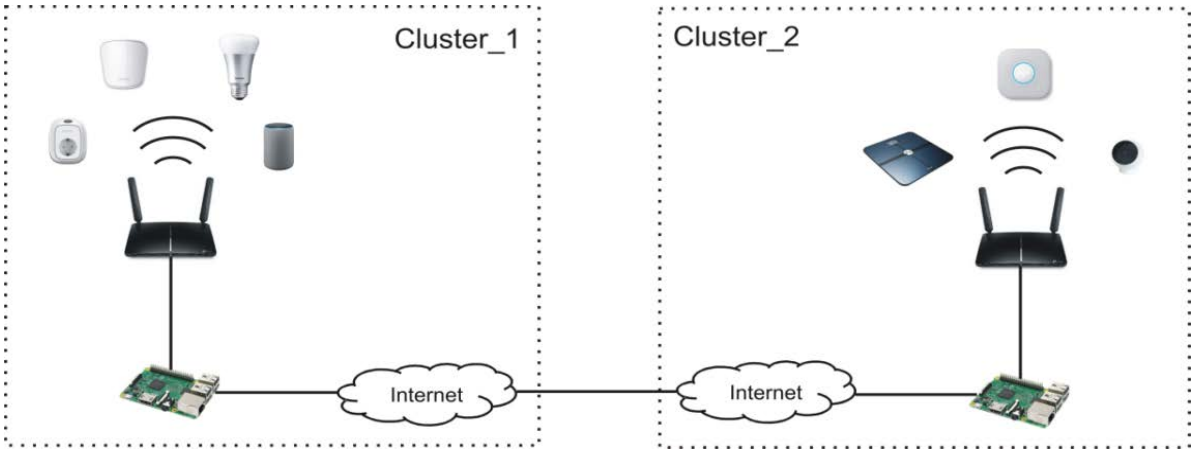

Figure 3: Social network testbed architecture

The duration of network traffic collection was 2 weeks each day at specified intervals. To automate the collection of network traffic, a script was written, whose startup planning was implemented using the Cron Job utility. To identify smart devices with the flow of network packets, we used the MAC address of the devices in the packet headers, which allowed us to differentiate the traffic of smart devices from each other. After receiving raw set of network traffic, it was converted to PDML format and features extracting was performed. As a result, a set of feature vectors was obtained that described the behavior of the packet sequence for the monitoring period *T*. These actions were repeated for both clusters.

Table 2
Placement of smart devices between clusters

№	Label	Smart device	Cluster	MAC address
1	A	Amazon echo	Cluster_1	44:65:0d:62:a4:d7
2	B	Belkin WeMo Switch	Cluster_1	ec:1a:59:a3:f1:4b
3	C	Belkin WeMo Motion	Cluster_1	ec:1a:59:d3:c0:17
4	D	Philips Hue Light Bulb	Cluster_1	00:17:88:28:45:81
5	E	NEST Smoke Sensor	Cluster_2	18:b4:30:35:f4:c3
6	F	TP-Link Camera	Cluster_2	f4:f2:6d:97:d8:10
7	G	Withings Scale	Cluster_2	00:24:e4:14:47:bd

During the experiment, we omitted the simulation of the first mode of operation of the system (monitoring of network traffic to detect abnormal activity, see Fig. 1). This is due to the fact that this mode can be easily implemented using an intrusion prevention system, such as Snort [36], by writing rules that will monitor the occurrence of abnormal activity. To activate the second mode, a Python script was implemented, which launched a profile search in the cluster and if such profile is absent, switched the system to the third mode – profile search in other social network clusters.

The first experiment involved determining the efficiency of identifying profiles of smart devices that are present in both the white list of profiles and outside this list locally on the same cluster (i.e., check the effectiveness of the second mode of the system in cluster 1).

To determine the effectiveness, a standard accuracy measure of classifier evaluation was used [37-40], which determines the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN), \quad (5)$$

The entire volume of data was divided into two parts: training (three quarters) and test sample (one quarter). The test sample included network traffic data from all four smart devices. The training sample was used to form a white list of behavioral profiles, i.e. to classify the classifier. In order to simulate abnormal activity, a total of 4 series of experiments were performed, in each of which the classifier was trained on data that included only three of the four smart devices (i.e. in each experiment only three smart devices were represented in the white list of profiles). Thus, a test sample containing the profiles of all smart devices in the cluster was used to verify the effectiveness of device identification. In each of the experiments, the threshold value of the probability δ that determines the belonging to the class of smart device d_i in the cluster c_j was chosen experimentally at the level of 0.56. The average results of four series of experiments are shown in Figure 4.

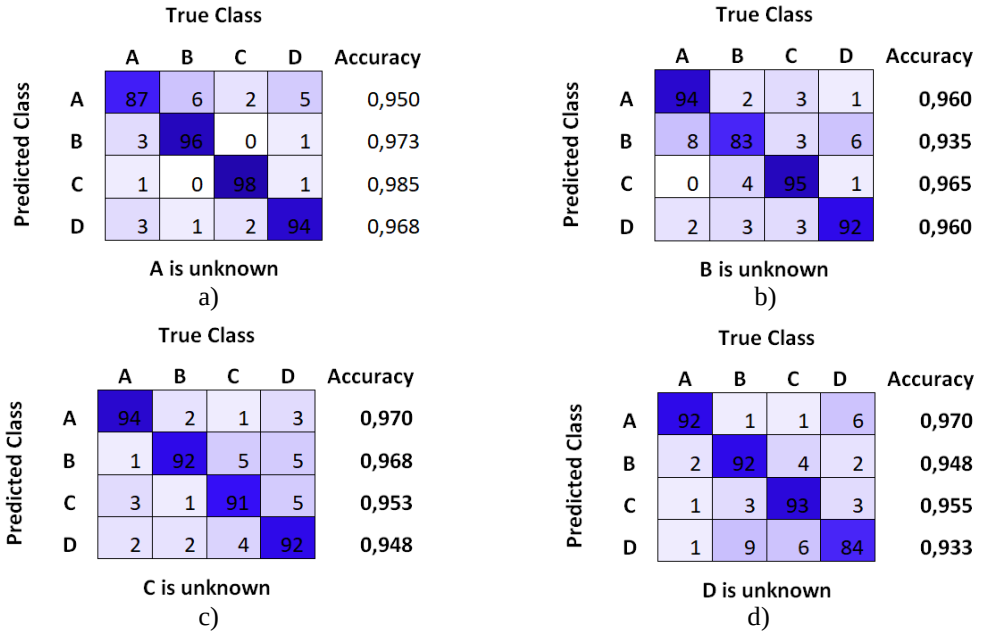

Figure 4: Confusion matrices for evaluating the accuracy of identifying profiles of smart devices that are present in both the white list of profiles and outside this list locally in Cluster_1:
a)Amazon echo outside white list; b) Belkin WeMo Switch outside white list;
c) Belkin WeMo Motion outside white list; d) Philips Hue Light Bulb outside white list.

The results of the experiments showed quite high results, in particular, the highest efficiency (0.985) was obtained in the first experiment, in the white list of which, there was no smart column Amazon echo. In this experiment, 98% of the network traffic sessions corresponding to the Belkin WeMo Motion sensor were identified as belonging to it (Fig. 4a true positive for class C). The lowest true positive value in the first experiment, as expected, was for class A (87%), i.e. when the system tried to predict Amazon echo data (which were outside white list). In the rest of the experiments, the average value of the accuracy of the system was 0.955 for the second experiment, 0.959 for the third and 0.951 for the fourth. It should also be noted that for experiments, the value of T was 20 sec, and the value t_i was 5 sec.

In the second experiment, the accuracy of the entire system was tested, i.e. the sequential execution of the second and third modes. The purpose of the experiment was to check whether the

system will be able to identify a device if this device is not in one cluster, but is in other clusters. For this purpose, two clusters were involved (Fig. 3) and white lists of device profiles present in it were created (the classifier was trained on training data consisting of smart devices present in this cluster). Three series of experiments were performed to identify the profiles of NEST Smoke Sensor, TP-Link Camera and Withings Scale (E, F, G, respectively) in cluster 1, the white list of which did not contain these smart devices. The results of the experiments are presented in table 3.

Table 3
Accuracy of detection of NEST Smoke Sensor, TP-Link Camera and Withings Scale in cluster_1

	E	F	G	Overall
The number of sessions	932	1420	502	
Accuracy, %	98,20	97,36	96,08	**97,21**
Sessions identified as A or B or C or D in Cluster_1 (False Positives), %	6,32	5,84	5,67	**5.94**

In this experiment, the False Positives level determined the number of sessions belonging to the test smart device (E, F and G), which were assigned to one of the devices marked as A, B, C and D. According to the results of the experiment, the overall accuracy of the system obtained at the level 97.21% with an average level of type I errors of 5.94%.

7. Conclusion

The paper proposes a new approach to the organization of security of smart homes, which involves their integration into clusters. To implement this approach, a system for detecting anomalies and identifying smart home devices using collective communication is presented. Detection of anomalies and identification of devices in each of the smart homes is based on monitoring network traffic and creating profiles of smart devices that are present in the network. Profiles consist of a set of features that describe the behavior of smart devices on the network, including the period of activity of the device and the period of its sleep. Based on this, a whitelist of allowed profiles of device operation in the cluster is formed. The Random Forest algorithm was used to check the presence of a profile in the cluster whitelist. If the observed profile is absent in the white list of the cluster, a request is made to other clusters that form a social network to compare the profile of the sequence of packets received in the cluster with their own whitelists. To evaluate the effectiveness of the proposed system, a number of experimental studies were conducted. The results of the experiments showed the overall accuracy of the system at the level of 97.21% with an average level of type I errors of 5.94%.

8. References

[1] O. Drozd, M. Kuznietsov, O. Martynyuk, M. Drozd, A method of the hidden faults elimination in FPGA projects for the critical applications. Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, Kyiv, Ukraine, 2018, pp. 231-234. doi: 10.1109/DESSERT.2018.8409131

[2] A. Melnyk, V. Melnyk, Self-configurable FPGA-based computer systems, Advances in Electrical and Computer Engineering (2013), 13(2), 33–38. doi: 10.4316/AECE.2013.02005

[3] O. Drozd, I. Perebeinos, O. Martynyuk, et al., Hidden fault analysis of FPGA projects for critical applications. Proceedings of the IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, Lviv-Slavsko, Ukraine, 2020. doi: 10.1109/TCSET49122.2020.235591

[4] O. Drozd, K. Zashcholkin, O. Martynyuk, et al., Development of Checkability in FPGA Components of Safety-Related Systems. CEUR Workshop Proceedings 2762 (2020) 30–42.

[5] A. Melnyk, V. Melnyk, Remote Synthesis of Computer Devices for FPGA-Based IoT Nodes, Proceedings of the 10th International Conference on Advanced Computer Information

Technologies, IEEE, Deggendorf, Germany, 2020, pp. 254-259. doi: 10.1109/ACIT49673.2020.9208882.
[6] C. Wheelus, X. Zhu, IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework, IoT 1 (2020) 259–285. doi: 10.3390/iot1020016
[7] E. Manziuk, W. Wójcik, O. Barmak I. et al., Approach to creating an ensemble on a hierarchy of clusters using model decisions correlation, Przegląd Elektrotechniczny (2020), 96(9), 108–113. doi:10.15199/48.2020.09.2
[8] I. Krak, O. Barmak, E. Manziuk Using visual analytics to develop human and machine-centric models: A review of approaches and proposed information technology, Computational Intelligence (2020), 1-26. doi: 10.1111/coin.12289
[9] G. Markowsky, O. Savenko, S. Lysenko et al., The Technique for Metamorphic Viruses' Detection Based on its Obfuscation Features Analysis, CEUR Workshop Proceedings, 2104 (2018) 680-687.
[10] R. Kitchin, M. Dodge, The (in)security of smart cities: vulnerabilities, risks, mitigation and prevention Journal of urban technology (2019), 26(2), 47-65. doi:10.1080/10630732.2017.1408002
[11] McAfee, Inc., McAfee Labs Threats Report: April 2017, URL: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2017.pdf
[12] E. Džaferović, A. Sokol, A. A. Almisreb, S. M. Norzeli, 1. DoS and DDoS vulnerability of IoT: A review, Sustainable Engineering and Innovation 1 (2019) 43–48. doi: 10.37868/sei.v1i1.36
[13] I. Ali, S. Sabir, Z. Ullah, Internet of Things Security, Device Authentication and Access Control: A Review, International Journal of Computer Science and Information Security 14 (2016) 456–466.
[14] A. Nicheporuk, O. Savenko, A. Nicheporuk, Y. Nicheporuk, An android malware detection method based on CNN mixed-data model, CEUR Workshop Proceedings 2732 (2020) 198–213.
[15] S. Lysenko, K. Bobrovnikova, A. Nicheporuk et al., SVM-based technique for mobile malware detection, CEUR Workshop Proceedings, 2353 (2019) 85-97.
[16] O. Pomorova, O. Savenko, S. Lysenko, et al., Metamorphic Viruses Detection Technique based on the Modified Emulators, CEUR Workshop Proceedings 1614 (2016) 375–383
[17] O. Savenko, A. Nicheporuk, I. Hurman, S. Lysenko, Dynamic signature-based malware detection technique based on API call tracing, CEUR Workshop Proceedings 2393 (2019) 633–643
[18] P. Spadaccino, F. Cuomo, Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Comuting, arXiv preprint arXiv:2012.01174 (2020).
[19] O. Savenko, S. Lysenko, A. Nicheporuk et al., Approach for the Unknown Metamorphic Virus Detection, Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IEEE, Bucharest, Romania, 2017, pp. 453-458. doi: 10.1109/IDAACS.2017.8095052
[20] A. Sivanathan, IoT Behavioral Monitoring via Network Traffic Analysis, Ph.D. thesis, The University of New South Wales, Sydney, Australia, 2020.
[21] O. Savenko, S. Lysenko, A. Nicheporuk et al., Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search, CEUR Workshop Proceedings, 1844 (2017) 555-569.
[22] I. Hafeez, M. Antikainen, A. Y. Ding, S. Tarkoma, IoT-KEEPER:Detecting malicious IoT network activity using online traffic analysisat the edge, IEEE Transactions on Network and Service Management (2020), 45–59. doi: 10.1109/TNSM.2020.2966951
[23] P. Dymora, M. Mazurek, Anomaly detection in iot communication network based on spectral analysis and hurst exponent, Applied Sciences 9, 5319 (2019). doi: 10.3390/app9245319
[24] D. H. Summerville, K. M. Zach and Y. Chen, Ultra-lightweight deep packet anomaly detection for internet of things devices, Proceedings of the 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), IEEE, Nanjing, China, 2015, pp. 1–8. doi: 10.1109/PCCC.2015.7410342
[25] I. Ullah, Q. H. Mahmoud, A Two-Level Flow-Based Anomalous Activity Detection System for IoT Networks, Electronics 9, 530 (2020). doi: 10.3390/electronics9030530
[26] S. Ramapatruni, S. N. Narayanan, S. Mittal, et al., Anomaly detection models for smart home security, Proceedings of the 2019 IEEE 5th Intl Conference on Big Data Security on Cloud

(BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, Washington, DC, USA, 2019, pp. 19-24. doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00015
[27] M. Yamauchi, Y. Ohsita, M. Murata, et al., Anomaly Detection in Smart Home Operation From User Behaviors and Home Conditions, Transaction and consumer electronics 66 (2020), 183–192. doi: 10.1109/ICCE.2019.8661976.
[28] A. Sivanathan, D. Sherratt, H. H. Gharakheili et al., Characterizing and classifying IoT traffic in smart cities and campuses, Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, Atlanta, GA, 2017, pp. 559–564. doi: 10.1109/INFCOMW.2017.8116438.
[29] J. Bugeja, A. Jacobsson and P. Davidsson, On Privacy and Security Challenges in Smart Connected Homes, Proceedings of the 2016 European Intelligence and Security Informatics Conference (EISIC), IEEE, Uppsala, Sweden, 2016, pp. 172-175, doi: 10.1109/EISIC.2016.044.
[30] S. Albishi, B. Soh, A. Ullah et al., Challenges and Solutions for Applications and Technologies in the Internet of Things, Proceedings of the 4th Information Systems International Conference 2017 (ISICO), Procedia Computer Science, Bali, Indonesia, 2017, pp. 608-614. doi:10.1016/j.procs.2017.12.196
[31] N. Apthorpe, D. Reisman, N. Feamster, A Smart Home is No Castle:Privacy Vulnerabilities of Encrypted IoT Traffic, arXiv preprint arXiv:1705.06805 (2016).
[32] L. Bedratyuk, O. Savenko, The Star Sequence and the General First Zagreb Index, MATCH Communications in Mathematical and in Computer Chemistry (2018), 79(2), 407-414.
[33] J. Bugeja, A. Jacobsson and P. Davidsson, On Privacy and Security Challenges in Smart Connected Homes, Proceedings of the 2016 European Intelligence and Security Informatics Conference (EISIC), IEEE, Uppsala, Sweden, 2016, pp. 172-175, doi: 10.1109/EISIC.2016.044.
[34] E. Anthi, L. Williams, M. Słowińska, et al., A Supervised Intrusion Detection System for Smart Home IoT Devices, IEEE Internet of Things Journal (2019), 9042–9053. doi: 10.1109/JIOT.2019.2926365
[35] Tshark – The Wireshark Network Analyzer URL: https://www.wireshark.org/docs/man-pages/tshark.html
[36] Snort, URL: https://www.snort.org/
[37] S. Lysenko, K. Bobrovnikova & O. Savenko, A botnet detection approach based on the clonal selection algorithm. In 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). IEEE (2018) 424-428.
[38] S. Lysenko, K. Bobrovnikova, S. Matiukh, I. Hurman & O. Savenko, Detection of the botnets' low-rate DDoS attacks based on self-similarity. International Journal of Electrical & Computer Engineering, 2020, 10, 2088-8708. DOI: http://doi.org/10.11591/ijece.v10i4.pp3651-3659.
[39] 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., Kwiecien A. (eds) Computer Networks. CN 2019. Communications in Computer and Information Science, vol 1039, p. 127-143. Springer, Cham (2019), doi: 10.1007/978-3-030-21952-9 10.
[40] Savenko O., Lysenko S., Kryschuk A. Multi-agent Based Approach of Botnet Detection in Computer Systems. Communications in Computer and Information Science, 2012, vol 291. pp 171-180. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31217-5_19.