ARCSECURE: Centralized Hub for Securing A Network of IoT Devices

Kavinga Yapa Abeywardena, A.M Isuru Srimal Abeykoon, A.M.S.P.B Atapattu, H.N Jayawardhane, C.N Samarasekara
Information System Engineering
(Cyber Security)
Sri Lanka Institue of Information Technology
Malabe, Sri Lanka
kavinga.y@sliit.lk, isuru.srimal258@gmail.com, supushpitha@live.com, helanij@gmail.com, chamath96@outlook.com

***Abstract*— With respect to current trends in information technology, Internet of Things (IoT)is playing a prominent role in technological advancements which has happened in the last few years. In the current context, the major issue that users face is the threat to their information stored in these devices. Modern day attackers are aware of vulnerabilities in the current IoT environment. Therefore, securing information from being gone into the hands of unauthorized parties is of highest priority for IoT users. With the need of securing the information came the need of protecting the devices which the data is being stored. Small Office/Home Office (SOHO) environments working with IoT devices are particularly in need of such mechanisms to protect the data and information that they hold in order to sustain their operations. Hence, to come up with a well-rounded security mechanism from every possible aspect, this research proposes a plug and play device called "ARCSECURE".**

Keywords— Internet of Things, Information Security, Machine Learning, DoS, DDoS, Botnet, Authentication, Authorization, Detection, Mitigation, Malware

I. INTRODUCTION

Protecting information, providing security for information is a major concern of any user. In a business environment, loss or unauthorized modification of information/ data could put the whole business's functions at a hold and even create great losses financially and reputationally. For a Small Office/ Home Office (SOHO) environment the organization will hold various information related to the organization, customers and suppliers etc. If in case one of this device are compromised the whole network of devices will be at a risk of being attacked. Therefore, each and every one of the devices within the network should be protected to guarantee that the network is safeguarded.

Malware attacks such as trojan horses and spyware could easily go undetected and the user could lose information without their knowledge. Many attacks go undetected due to the lack of expertise within the users to detect suspicious behavior and in some cases even if the user was able to identify anomalies in the functionality of the system it can be hard to come up with a mitigation option without the proper knowledge and expertise because simply shutting down the machine won't help in this case. Therefore, it is essential that better mechanisms are put into place to immediately detect and address such attacks.

Even if such attacks are detected and stopped from entering the system at an early stage, there can be instances where authentication stage is compromised by ways such as brute forcing. Hence all possibilities should be considered when coming up with a well-rounded security solution.

Another aspect that does not gain as much attention is the initial stage when a user will be installing and configuring the relevant software for the devices that they have purchased. In this often a user might be directed to a cracked version of the software where the user will believe it to be legitimate, yet it might be a file with malware such as ransomware. If the user downloads this executable and runs it the whole system will be compromised, and it might result in the user losing all his/her important data where in a business environment this could be crucial.

Therefore, it can be said that taking all abovementioned aspects into account is a necessity when coming up with a solution to protect business functionalities in a SOHO environment working with a network of IoT devices. Hence the device proposed by this research "ARCSECURE" will be addressing all these issues in order to come up with a conversant solution which will be ideal for a SOHO environment.

II. LITERATURE REVIEW & RELATED WORK

Stepping into the future of technology, IoT, as suggested by Haller et al, "A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business process" carry many security problems [1] First and foremost amongst the issues are, vulnerabilities produced by poor designing of the program, which will in return create backdoor installations and malware insertion opportunities for attackers. [1] Overlooking these issues in security has the ability to compromise the availability [1] as well as integrity and confidentiality of "IoT". As discussed by the Open Web Application Security Project (OWASP), insecure software/firmware comes under top 10 vulnerabilities identified for the architecture of IoT [2]. There can exist instances where an attacker could disguise an old version of software which contains security vulnerability as the version offered as the latest by the vendor[3]. This would revert the system back to a faulty version giving attacker access[3]. A faulty software could leave the hardware vulnerable involuntarily[4].

When it comes to authentication and authorization keystroke dynamics have been researched for a long time. In 2002 Bergadano et al [5] researched keystroke dynamics for user authentication using the volunteer's self-collected dataset, data collected using the same text for all individuals, resulting in only 0.01 percent passing the authentication impostor. In 2003, Yu and Cho [6] performed preliminary experimental research on the selection of a function subset selection of keystroke dynamics identity verification and found that GA-SVM yielded good accuracy and speed of learning. Revett et al. [7] researched user authentication in 2007 and began researching authentication for a dynamic keystroke. The author has suggested that biometrics are robust, specifically fingerprints, but they can be easily spoofed.

In their research paper Shivaramu K and Prasobh P.S discuss Security Vulnerabilities that the Wireless Ad-hoc network is vulnerable due to cooperative algorithms, lack of monitoring and management and lack of a particular line of defense [7]. An important conclusion drawn from the experimental results is that, of the various methods used, Fuzzy c-means clustering is very efficient in detecting DDoS attacks [8].

Going into another type of prominent attack types, it is stated that recent botnet attacks are more likely to occur in IoT systems and devices than other traditional network environments. From the basis of this research, literature has been reviewed. Article [9] explains that common IoT vulnerabilities such as, web, cloud or mobile interfaces with insecure interfaces, authorization and authentication methods which aren't sufficient, network services which are insecure, software and firmware which consists vulnerabilities etc.

III. METHODOLOGY

A. Detection of IOT Botnet Attacks

This project involves building a Host-Based IDS for SOHO Networks. Host-Based IDS is divided into two subcategories, as Packet Capture module and Botnet Prevention System (BPS).

1) Packet Capture Module

Packet Capture module contains three sub modules:

- **Packet Capturer:** used to read data of the incoming packets. Libraries: pyshark, afpacket and tshark (Wireshark API). When the new packet comes into the network packet capturer gather the raw data and send the data to packet parser.
- **Packet Parser:** Used to retrieve the meta data out of the raw data that came from the packet capturer. Libraries: tshark.
- **Feature Extractor (FE):** this module is used to extract *n* features from the incoming packets to create the *x* instance ($x_i \in \mathbb{R}$). The *x* instance holds the attributes of the packet. The *n* features that contains 115 traffic statistics.

For network anomaly detection, it is important to extract features from every packet that is coming through the packet parser. Hence, authors implemented a solution for high speed feature extraction. This method uses *O(l)* complexity because of the incremental statistics. These incremental statics maintained over a damped window (only capture the recent behaviors). This method also uses 2D statistics which helps to the connection between the rx and tx network traffic. Also, this method consists with the memory requirement and runtime complexity of *O(n).* Because of the runtime complexity the weight of the older values is decreased over time. The key features extracted include,

1. **SrcMAC-IP:** MAC and IP address of the packet source
2. **SrcIP:** IP address of the Source
3. **Channel:** Channel used between sender and receiver
4. **Socket:** Socket of the packet sender and receiver (TCP/UDP)

FE extracts 23 features from single time frame. Time frames are 1min, 10sec, 1.5sec, 500ms, and 100ms this totaling 115 features.

Table 1

PACKET	FEATURES EXTRACTED	#FERTURES
SIZE	SrcMAC-IP, Src-IP, Channel, Socket	8
SIZE	Channel, Socket	8
COUNT	SrcMAC-IP, Src-IP, Channel, Socket	4
JITTER	Channel	3

2) Botnet Prevention System (BPS)

BPS act as the core system for mitigating botnet attacks. This is based on a machine learning model, a Feature Mapper (FM) and Anomaly Detection (AD) modules.

a) Machine Learning Model

For botnet detection machine learning model is divided in to two parts,

- **Data Analytics**
  Data analysis helps to create better model from the raw data. That's make easier the machine learning process and it can help to quantify and track objectives.
- **Machine Learning Algorithms**
  The data set that used in here UCI Dataset [10]. This dataset contains 7062606 instances and 115 real attributes. Below are the accuracy results obtained based on each machine learning algorithm.

Table 2

Algorithm	Accuracy
Decision Tree	98%
Logistic Regression	99.98%
Perceptron	98.73%
Naïve Bayes	97%
K-Nearest Neighbours (KNN)	99.97%
Decision Tree	98%

b) Feature Mapper (FM)

This module is responsible for mapping the *n* features that are collected through the FE to a vector(*x*). Different smaller sub instances*(k)*, one *k* for every encoder in the ensemble layer of Anomaly detector. Let v denote the set of *k*,

$$v = \{v_1, v_2, v_3, \dots, v_k\}$$

FM consists of two distinctive modes,

- **Train-Mode:** using input vector *x* the model learns a feature map.

- **Execution-Mode:** the learned mapping is used to create a collection of small instances of *v* vector from *x* vector.

c) Anomaly Detection (AD)

- **Train-Mode:** It takes the v to train the respective Autoencoder in the ensemble layer. Now the RMSE is calculated during the forward-propagation and it is used in training the output layer.
- **Execution-Mode:** The v vectors derived from the execution phase of the Feature Mapper is executed at the respective autoencoders of the ensemble.

$$RMSE\left(\underset{x}{\rightarrow},\underset{y}{\rightarrow}\right)=\sqrt{\frac{\sum_{i=1}^{n}(x_i-y_i)^2}{n}}$$

x is capability to recreate hidden instances from the same data distribution *where*,
$\underset{x}{\rightarrow}$ is reconstruction error of the instance.
$\underset{y}{\rightarrow}$ is reconstructed output.

The AD consists with two layers of Autoencoder,

- **Ensemble Layer:** This measures the independent abnormality of each sub-instance in *v*.
- **Output Layer:** This layer output the final anomaly score by considering abnormalities of the instance and noise in the network.

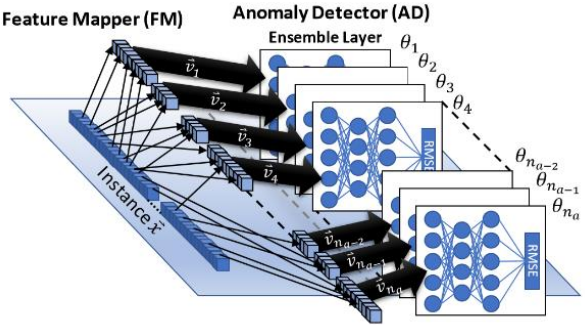

Figure 1: Mapping Process

B. Malware Detection and Mitigation

1) Detection

a) Feature extraction

Machine Learning will only use integer or float data types as features when it comes to detection. Data such as "*entropy*" is vital when it comes to detecting malware. 'Entropy' refers to randomness or messiness of a code. It is stated that if the entropy is high (high messiness/randomness) then the possibility of the code carrying malware would be high. Hence, the minimum, mean and maximum values of entropy will be extracted to determine if the file is malicious or not. All Portable Executable (PE) features were extracted with use of *"pefile"* reader module in Python library.

b) Feature Selection

Feature selection for this model was done with use of the Tree-based feature selection of SciKit library. Hence the original data of which contained 54 columns/features will be narrowed down to 14 most important features to reduce dimensionality of the dataset. The final features extracted were as follows,
The original dataset was then split in to two as 80% of the original dataset as training data and the remaining 20% of testing data.

c) Selection of Classification Algorithms and Training

Subsequently, by taking the training dataset into consideration the model was trained with 5 different algorithms with the goal of choosing the best possible algorithm that matches the requirement. Table 3 depicts the algorithms which were used, and results yielded.

Table 3

Algorithm	Accuracy
Decision Tree	98%
Random Forest	99%
Gradient Boosting	98%
AdaBoosting	97%
GNB	70%

From the above table it is evident that Random Forest algorithm gave the best result in detecting malicious content. Therefore, Random Forest algorithm was taken as the most suited algorithm and henceforth used to train the data for this module.

d) Testing

In the testing phase the entropy of the input file will be calculated first. The calculation will be as shown below.

$$p(x) = Pr\{X = x\}, x \in X$$

Let x be a discrete random variable with alphabet X and probability of p(x)

Then entropy H(X),

$$H(X) = \sum_{x \in X} p(x)\, logp(x)$$

(The base for the logarithm in this module was used as 2)

Then the version information and other information such as size of code, base of code etc. will be taken into consideration to see if the given file's values tallies with the legitimate file's mean, min and max entropy values and other values that was taken into consideration. By giving few entries (a mix of both malicious and legitimate files) from the testing dataset as inputs it can be confirmed that values were predicted as "legitimate" and "malicious" with a high level of precision.

2) Mitigation

The next step of the module is implementing the ML model in the web environment where the user interaction with the system will be taking place.

When the files are inserted to the web application's malware detection module, the malicious executable will be identified as malicious and the ARCSECURE site will display a message saying that the file has been identified to be malicious. In these cases, the user will be prompted to either delete the file from the system or proceed with the installation.
Subsequently, ARCSECURE will maintain a log of the files that have been recognized as "malicious" and "legitimate" separately in order to prevent the user from installing malicious content in the future.

C. DoS Detection and Mitigation Module

Detecting DDoS threats to utilize and communicate IoT devices with a proper manner is the main purpose of this module.

1) Attack Detection Module

All incoming traffic will be going through this module to complete the scanning process of this function and this module contains a signature and a policy engine. Attacks of DDoS can be identified by inspecting the system traffic changes. In a study performed by Chonka et al. [11], by utilizing the property of system self-similitude a model is created to discover DDoS flooding traffic. The centralized node is controlling the boundaries of detectors. The objective of this intrusion detection framework is to upgrade the general execution of DDoS attack recognition, by shortening the discovery delay while expanding the detection accuracy and seed of the system communication.

2) Data Training Module

In this module authors train the machine using data which have been gathered for training purposes and the second approach is to test the captured data. Well trained ML algorithm will clearly classify the abnormal patterns from normal data packets. It will easily identify any anomalies inside the IoT environment.

3) Mitigation Module

This will be the final module in the framework and all incoming network traffic will also be going through here for IoT devices or if the purposed system will be able to detect any anomalies the packets are dropped or if they classify into safe category they will be forwarded into IoT environment.

a) Machine Learning for DDoS detection

Networks always face challenges in distinguishing legitimate and malicious network packets. Random forest ML algorithm operates by constructing multitude decision trees at training time, Logistic regression is a statistical model and more complex extensions exists on it [12]. A major drawback on standard feedforward exist where information moves only on one direction and there is no cycles or loops in this neural network algorithm. Support vector machine (SVM) uses classification algorithms for 2 group classification problems. SVM is successful when classifying datapoints into their corresponding classes.

b) Support Vector Machine

Support Vector Machines (SVMs) are learning machines that plot the preparation vectors in high dimensional component space, marking every vector by its group. SVM classifiers give a component to fit a hyper plane to play out a linear classification of the patterns using a kernel function. Support vector machine do not require a decrease in the quantity of highlights so as to maintain a strategic distance from over fitting-an obvious preferred position in applications. Another essential preferred position of SVMs is the low expected probability of generalization errors.

c) Detecting DoS Attacks using SVMs

To assemble SVMs for DoS discovery, the input vectors are extricated from raw network packet dumps in the test data set; the outcome is dependent on the SVM preparing boundaries and they are as follows,

i. Speed of source IP (SSIP)
ii. Standard Deviation of incoming flow packets (SDFP)
iii. Standard Deviation of incoming flow bytes (SDFB)
iv. Speed of flow entries (SFE)
v. Ratio of pair-flow entries (RFIP)

The results of the above parameters of SVM is trained on is shown in the figure 3.

A stream sections characterized as an interactive flow if there is bidirectional communication between the source and the destination in the stream.

The standard deviation of flow packets and bytes decreases because of the high number of packets resulting the expanding number of flows. However, there is an extremely slight variation in the packet size just as the byte size which brings about lower deviation. Ratio pair of stream entries decreases the DoS attack situation since the reaction set out by the objective machine is missed out in the address space. As shown in the figure 3, there is an expansion during DoS attack for the speed of source IP address since we are utilizing IP spoofing procedure to re-enact the DoS attacks.

D. Secure User Authentication

1) Keystroke Dynamics

User authentication is one of the most vulnerable areas of compromise in any system [13][14]. Therefore, a robust mechanism should be in place to secure access only to legitimate users. The proposed solution to strengthen the user authentication process involves the use of Keystroke dynamics. The use of behavioral biometrics such as Keystroke dynamics for authentication purposes are not unheard of. However, despite their increasing use in various other fields their usage in the field of IoT is yet to materialize [15][16].

2) Dataset

The dataset which was chosen for the supervised model was a very commonly used standard dataset for keystroke dynamics which was readily available on the web. This dataset included the timings of various key holds recorded in milliseconds.

3) Feature Extraction

The dataset contains more than 100 attributes of typing timings. Out of which the following features were taken into consideration for the implementation of this model.

- **Hold time** – the time interval between the press of a key and release of the same key.
- **Down-Down time** – the time interval between the press of a key, release of the same key and the press of the next key.
- **Up-Down time** – the time interval between the release of a pressed key and the press of the next key.

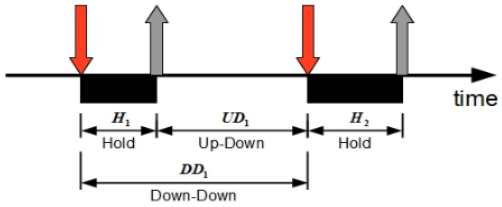

Figure 2: Keystroke Data Collection Features

4) The Model

Keystroke dynamics involved the use of machine learning to train a supervised model that should be trained for a specified number of times. The model used was based on a neural network known as Long Short-Term Memory (LSTM) network. A LSTM network is a type of Recurrent Neural Network (RNN) [17]. The model to be chosen for Keystroke dynamics had to address two essentials which are long term retention of captured keystrokes and the ability to reason on previous occurrences to decide on future ones. Normal neural networks are incapable of retaining past information for long periods therefore, RNN's were designed to satisfy this purpose by having loops which enabled past data to persist [17]. However, there was a drawback RNN's struggled to learn from past information to predict future outcomes [17]. This was accomplished by LSTM networks, therefore, was chosen as the model to implement Keystroke dynamics. The model was designed by splitting the dataset into two halves as testing and training sets, respectively. The training set was used to train the algorithm. Then the testing set was used to test the algorithm and evaluate its accuracy.

5) Testing

A simple dummy web application was developed for testing purposes using basic front-end development programming languages such as HTML, CSS, and JavaScript. The machine learning module was then exported and integrated with the web application's user input fields. Eventually, upon the user entering their password, the model would compare the typing pattern to that of the trained patterns. If the users typing characteristics manages to match or fall in-between the average range recorded during the users testing phase, then that user is deemed legitimate and thereby authenticated to access the system. If not, the user is deemed an imposter and would not be allowed to progress any further.

IV. Results And Discussions

1) Detection of IOT Botnet Attacks

In summation, the machine learning techniques use to perform evaluations on the data set. As an outcome autoencoders were able to train successfully to detect anomalies based on the anomaly score estimate. Since the Output layer learns the patterns and relations between the input features in the training data, the Ensemble layer's return the anomalies in the data and differentiate malicious instances of data from the benign instances. The algorithm was able to successfully detect anomalies in the cross validation and test data set.

2) Malware Detection and Mitigation

As per the results of the Malware detection and mitigation module of "ARCSECURE" Random Forest algorithm was taken as the most suitable algorithm to train the model and with the use of that the model was able to successfully determine malware files from legitimate files. With the file log in place the detection module will enable the user to delete the malicious files and proceed with the correct installation process.

3) DoS Detection and Mitigation Module

Support Vector Machines (SVM) effectively accomplish high detection accuracy (more than 99%) for each DoS attack [18] instances of data.

There is another IP which is entering the system at each moment, so the module needs to handle this issue and make a new flow entry for every single IP address entering the pool. that outcomes in increment number of flows per unit time. SFE diagram additionally has an expansion during DoS attack since more number of flows are made into the module to handle the enormous number of incoming IPs.

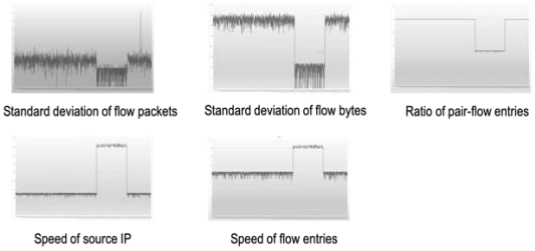

Figure 3: Results of SVM training parameters

The above parameters and the following figure 4 show the result of SVM classifier.

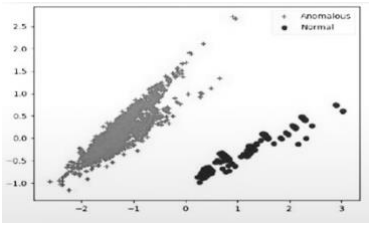

Figure 4: Results from SVM

Figure 4 shows that the highlights of high correlation with the event of an attack and dependent on the highlights separated. It is evident that SVM is successful in grouping data points into their corresponding classes.

4) Secure User Authentication

The chosen model yielded an accuracy level of about 85%. It was observed that increasing the value for the number of epochs did not have a significant effect on the accuracy level. Epochs in machine learning refers to the number of cycles a

training dataset of a model has completed. The general consensus would be that the higher the number of epochs the better the accuracy which is not the case as there was a slight increase in accuracy when increasing the epoch from 5 to 10 but from there onwards the accuracy stayed more or less the same upon further increase.

V. CONCLUSIONS & FUTURE WORK

Internet of things presents a plethora of opportunities and avenues for innovation, However everyday IoT users might not realize the degree of protection and security risks associated with this technology. For that matter any device that shares a wireless connection faces the risk of security breach of one kind or the other. It is evident from the functions and features of 'ARCSECURE' presented in this paper that it successfully achieves and outperforms current applications in the niche market of local IoT security products.

Most of the time organizations have to pay a considerable amount of money to hire an expert to configure these devices and it takes more time to get the final usability of that device. The proposed device 'ARCSECURE' will be able to adopt to the network and do the needed configurations by itself. The product can offer maximum security for devices which are not capable of having inbuilt security mechanisms due to their various limitations. Proposed solution is capable of blocking malware, password compromises, identity theft and denial of service attempts and many more while delivering a high level of performance. Even without a thorough knowledge in IoT, the users will be able to experience high productivity through the proposed device. Despite unsupervised and supervised machine learning techniques are used in the proposed solution, ARCSECURE manages to maintain the all-important usability features of an IoT network.

As per the future work, the final outcome of the research will be to have a fully functional integrated system which will have the ability to detect and mitigate attacks such as botnets and DDoS, detect and mitigate malware, and authenticate and authorize legitimate personnel. This research is also expected to aid in building systems related to this context in future research.

REFERENCES

[1] Z. Zhang et al., "IoT Security : Ongoing Challenges and Research Opportunities," 2014.

[2] M. Ahmad and K. Salah, "IoT security : Review , blockchain solutions , and open challenges," Futur. Gener. Comput. Syst., vol. 82, pp. 395–411, 2018.

[3] H. Lin and N. W. Bergmann, "IoT Privacy and Security Challenges for Smart Home Environments," 2016.

[4] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," Proc. Asia South Pacific Des. Autom. Conf. ASP-DAC, vol. 25-28-Janu, pp. 519–524, 2016.

[5] G. Jonsdottir, D. Wood and R. Doshi, "IoT network monitor," 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, 2017, pp. 1-5. [Accessed 22 02 2020].

[6] Yu, E., Cho, S. Ga-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. In: Neural Networks, 2003. Proceedings of the International Joint Conference on; vol. 3. IEEE; 2003, p. 2253–2257.

[7] Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., Magalhaes, S., Santos, H.. A machine learning approach to keystroke dynamics-based user authentication. International Journal of Electronic Security and Digital Forensics 2007;1(1):55–70.

[8] Zahid, S., Shahzad, M., Khayam, S.A., Farooq, M.. Keystroke-based user identification on smartphones. In: International Workshop on Recent Advances in Intrusion Detection. Springer; 2009, p. 224–243.

[9] MooseFS, "moosefs.com," MooseFS, 03 2019. [Online]. Available: https://moosefs.com/. [Accessed 20 Feb 2020].

[10] The UCI dataset of botnet traffic [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/00442/. [Accessed 20 Feb 2020].

[11] A. Chonka, J. Singh, and W. Zhou, "Chaos theory based detection against network mimicking DDoS attacks," IEEE Communications Letters, vol. 13, no. 9, pp. 717–719, 2009.

[12] John O. Rawlings, Sastry G. Pantula, David A. Dickey, "Applied Regression Analysis", Springer.

[13] El-Hajj, M., Fadlallah, A., Chamoun, M., & Serhrouchni, A. (2019). A survey of internet of things (IoT) authentication schemes. Sensors (Switzerland), 19(5), 1–43. https://doi.org/10.3390/s19051141

[14] Muliono, Y., Ham, H., & Darmawan, D. (2018). Keystroke Dynamic Classification using Machine Learning for Password Authorization. Procedia Computer Science, 135, 564–569. https://doi.org/10.1016/j.procs.2018.08.209

[15] Geneiatakis, Dimitris & Kounelis, Ioannis & Neisse, Ricardo & Nai Fovino, Igor & Steri, Gary & Baldini, Gianmarco. (2017). Security and privacy issues for an IoT based smart home. 10.23919/MIPRO.2017.7973622.

[16] Dean, Andrew & Opoku Agyeman, Michael. (2018). A Study of the Advances in IoT Security. 1-5. 10.1145/3284557.3284560.

[17] *Understanding LSTM Networks -- colah's blog* (pp. 1–8). (2016). http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[18] Srinivas Mukkamala', Andrew H. Sung, "Detecting Denial of Service Attacks Using Support Vector Machines ," Institute for Complex Additive Systems Analysis New Mexico Tech Socorro, The IEEE International Conference on Fuzzy Systems