2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-0424-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/PERCOMWORKSHOPS51409.2021.9430966

Intrusion Detection System for Bluetooth Mesh Networks: Data Gathering and Experimental Evaluations

Andrea Lacava, Emanuele Giacomini, Francesco D'Alterio, Francesca Cuomo
Sapienza University of Rome, 00184 Rome, Italy
Email: {lacava.1663286, giacomini.1743995}@studenti.uniroma1.it
Email: {francesco.dalterio, francesca.cuomo}@uniroma1.it

***Abstract*—Bluetooth Low Energy mesh networks are emerging as new standard of short burst communications. While security of the messages is guaranteed thought standard encryption techniques, little has been done in terms of actively protecting the overall network in case of attacks aiming to undermine its integrity. Although many network analysis and risk mitigation techniques are currently available, they require considerable amounts of data coming from both legitimate and attack scenarios to sufficiently discriminate among them, which often turns into the requirement of a complete description of the traffic flowing through the network. Furthermore, there are no publicly available datasets to this extent for BLE mesh networks, due most to the novelty of the standard and to the absence of specific implementation tools.**

To create a reliable mechanism of network analysis suited for BLE in this paper we propose a machine learning Intrusion Detection System (IDS) based on pattern classification and recognition of the most classical denial of service attacks affecting this kind of networks, working on a single internal node, thus requiring a small amount of information to operate.

Moreover, in order to overcome the gap created by the absence of data, we present our data collection system based on ESP32 that allowed the collection of the packets from the Network and the Model layers of the BLE Mesh stack, together with a set of experiments conducted to get the necessary data to train the IDS.

In the last part, we describe some preliminary results obtained by the experimental setups, focusing on its strengths, as well as on the aspects where further analysis is required, hence proposing some improvements of the classification model as future work.

***Index Terms*—Bluetooth, BLE Mesh, Intrusion Detection System, IoT, network security**

I. Introduction

The approaching Internet of Things era will lead billions of entities to be simultaneously connected, opening new challenges about network management and data exchange rules. While the goal of IoT to realize an environment within which things are uniquely identified and able to interact with one another through the exchange of information seems really close, the concerns about security and privacy still remain far to be solved.

In fact, the exponential increase of the number of agents in the network determines an equivalent increase in vulnerabilities that can be exploited by criminals to gain access and control of the systems. In addition, the use of technologies that have not yet followed an official standardization process and therefore do not have full support from open source communities, makes it even more urgent to create defense systems specifically designed to act in conditions where energy consumption is a concern and where there is no use of IP stacks, such as IoT mesh intranets.

This paper proposes an Intrusion Detection System (IDS) specifically designed for Bluetooth Low Energy (BLE) Mesh networks which performs a pattern classification based on the analysis of the traffic flow.

The main reason of this choice lies in the fact that BLE Mesh is one of the leading protocols for nearby wireless communications with a structured standard recognized by the Bluetooth SIG family[1] and, while its security has been designed to be internal to the protocol, it still lacks of more complex network analyzers like Intrusion Detection Systems, able to promptly detect network anomalies.

Therefore, starting from these concepts, a classic pattern analysis model is implemented in a completely new communications field and it is demonstrated that the transmission parameters of new generation wireless networks such as BLE Mesh are valid enough to allow a classification of the network behavior. Consequently, instead of using network simulators to train and validate the obtained model, an innovative distributed data capture system has been created to allow in a consistent way the analysis of packet traffic generated by real IoT devices.

While the pattern classification approach is quite common in various IDS implementations, systems based on data gathered by a BLE Mesh network flow have never been designed. This gap is mainly attributable to the existence of just one published dataset [2], containing only packets without any malicious activity while a lot of different datasets with different behaviours are required to enable the classification of the network activities. These datasets have been experimentally generated with a real IoT testbed platform and an agnostic data collection framework created to extract the packets from the devices Both the IoT testbed and the *Data collector* will be presented in Section IV, along with the various configurations of the experiments, from a fully legitimate network routine to a network with a relay attacker capable of dropping all the

incoming packets.

After an overview of the security in BLE mesh networks in Section II, we present in Section III the Intrusion Detection System from a theoretical point of view, focusing on both its work phases.

Then in Section IV, we describe the experimental setup by introducing our reference platform, the features used in this work to acquire the data of the network traffic and the various scenarios implemented to collect the different types of packets needed.

In Section V we analyze the datasets acquired discussing its strengths and weaknesses and how these data can be used in different future approaches. Lastly, the performance results of the Intrusion Detection System proposed are discussed and compared among the different scenarios implemented.

II. Security in BLE Mesh networks

Bluetooth technology has been designed to be the standard for secure short range wireless connectivity. Bluetooth is a point-to-point wireless communication protocol operating in the unlicensed 2.4GHz ISM band, sharing it with other technologies such as WiFi protocol (IEEE 802.11) or ZigBee (and other implementations of IEEE 802.15.4). In 2019, the SIG Alliance formally decided to standardize the support of mesh networking using the BLE stack as a building base. The new mesh capability enables many-to-many device communications and is optimized for creating large-scale device networks that are suited for building automation, sensor networks, asset tracking, and any solution that requires tens, hundreds or thousands of devices to reliably and securely communicate with one another.

Any standard-compliant BLE Mesh strictly requires the security of the messages to be granted in any moment of the mesh network life cycle. At the beginning of the network formation phase, this requirement is enfoced by the device responsible of the creation of the network, called **provisioner**, which adds the other devices. Every mesh network must have at least one Provisioner in order to be formed.

All mesh communications are secured using AES-CCM with 128-bit keys, ensuring all mesh messages are encrypted and authenticated. There are various encryption keys that are used inside mesh communications.

The first one is the **Device Key (DevKey)**, which is a special application key that is unique to each node. Such key is used to encrypt the messages transferred between the Provisioner and the node when the device is provisioned and configured.

Then, in order to secure communications at the upper transport layer, an **Application Key (AppKey)** must be used for decryption of application data before delivering them to the application layer.

Finally, **Master Security Material** is derived from the **network key (NetKey)** and can be used by other nodes in the same network. Messages encrypted with master security material can be decoded by any node in the same network.

In order to prevent replay attacks, the BLE Mesh standard makes use of a sequence number *SEQ*. Every device increases the sequence number for each new message sent. Since sequence numbers are limited, a 32 bit **Initialization Vector (IV) Index** is used to overcome this bound and must be keep updated through the time, which is guaranteed to be unique for all the nodes in the same mesh network.

In complex networks of IoT devices often happens that devices can have multiple sensors in turn with multiple features enabled, therefore some nodes are more complicated than others and consist of multiple independent parts called **Elements** built on top of their roles. Each BLE Mesh node has at least one element, known as the **primary element** responsible of the configuration of the device, and may have additional elements that fulfill purpose of the network.

Elements are composed of entities that define a nodes' functionality and the condition of the element. These entities are called **Models** and are composed of **States**. Models define and implement the functionality and behavior of a node, while states define the condition of elements. Bluetooth mesh supports composite states, which are states composed of two or more values. A color changing lamp is an example, as the hue may change independently of color saturation or brightness.

Every model has a unique identifier, making it uniquely addressable.

Using messages, a Bluetooth mesh network communicates at the higher level via a client-server architecture. The function of a server is to expose the states of an element of which the client can read and modify the value.

Despite the proposed security mechanisms, Bluetooth Low Energy Mesh networks are still vulnerable to attacks. Even if there is a wide number of different keys, most of them are based on the device key and on the application key, which can both be recovered by hardware exploitation [3], allowing the attacker to take the full control of the target device without any secure fallback procedure from the protocol standard point of view.

One of the most severe threats affecting the BLE Mesh network is embodied by the *selective forwarding attack* [4], where a malicious relying node does not forward selected messages, thus hindering their propagation. This behaviour is called a *Gray hole* attack if it affects only a subset of the messages while in extreme cases of complete denial is called *Black hole* attack. Due to flooding approach for implementing many-to-many communication in BLE mesh networks, the effectiveness of this attack depends on the number of infected nodes, the density of the network and on the roles the nodes have within the network.

III. IDS architecture

We propose an Intrusion Detection System that bases its signature analysis on machine learning, aiming to understand the presence of attackers within the network. This research focused mostly on the *Black Hole* and the *Grey Hole* attacks since they are the simplest ones to be implemented yet very disruptive because a single node affected can alter or totally block all the communications of some portion of the network. Moreover, the IDS has the possibility to be extended with new

attacks on the condition of possessing a dataset that contains their trace.

The architecture of the IDS proposed is shown in Figure 1. The entire process logic is performed by an internal node of the system, named *watchdog*, which is able to analyze and classify all the traffic passing through, thus detecting the threats indicated above. The operations performed by the watchdog can be summarized in two main phases: the *learning phase*, where the classifier model is created by training a shallow neural network, and the *detection phase*, where the model is actually deployed over the BLE network to protect.

A. Learning phase

The most critical phase in any IDS' duty-cycle is represented by the learning phase, in which the system needs to acquire some information regarding the network to monitor, trying to find traffic patterns which can be used to successively discern among legit traffic flows from the ones containing malicious activities.

For supervised-learning IDS systems, this phase is generally composed of two main steps: the **preprocessing** and the **training**. In the first one, the aforementioned amount of information is extracted from traffic logs, readily collected beforehand or generally coming from external sources such as available datasets. These data are then properly filtered and manipulated to avoid biased configurations and to maximize their capacity to represent the attack scenarios into account. In particular, a set of needed features is extracted to create the final detection model. Since our work is based on the analysis of patterns in both legit and attack scenarios, hence requiring a suitable dataset not currently available in literature, we generated some specific traffic to obtain the needed datasets, using the testbed described in the next section.

Such data is divided into a list of **time windows** τ_i, each one with a prefixed duration t expressed in seconds and then a set of parameters is extracted: size of the time window n_i intended as the number of packets in it, average rssi $\mu_{RSSI,i}$, standard deviation of rssi $\sigma_{RSSI,i}$, average time to live $\mu_{TTL,i}$, standard deviation of the time to live $\sigma_{TTL,i}$, average number of packets per source address $\mu_{SRC,i}$, standard deviation of number packets per source number $\sigma_{SRC,i}$, $\mu_{DEST,i}$ average number of packets per destination address, $\sigma_{DEST,i}$ standard deviation of number packets per destination number. Furthermore, each feature of the time window (1) is scaled according to the L2 normalization strategy.

$$\tau_i = (n_i, \mu_{RSSI,i}, \sigma_{RSSI,i}, \mu_{TTL,i}, \sigma_{TTL,i}, \mu_{SRC,i}, \sigma_{SRC,i}, \mu_{DEST,i}, \sigma_{DEST,i}, y_k) \quad (1)$$

The training step also requires to analyze the collected data and label them. Each window (or sample) has been marked with two labels The first determines in binary form whether the sample contains malicious traces (**Attack**) or not (**Legit**). The second label also makes a distinction between the type of attack, such as Gray Hole (**TargetedGreyHoleAttack**), Black Hole (**BlackHoleAttack**) attacks and legitimate (**Legit**) traffic flow.

The dataset is further divided into 70/30 partitions for training and validation respectively. Samples coming from the partitions, are then batched in groups of 32 samples while the training procedure lasts 3000 epochs. To prevent over fitting, an early stopping technique is further used. The optimization procedure is carried on with an Adam optimizer [5] with a learning rate of 1×10^{-3}, which is an improvement of the well-known stochastic gradient descent algorithm for solving non-convex optimization problems. It has been chosen for its capability to perform well in finding an approximate solution for this problem even with a constrained amount of memory, CPU and time.

B. Detection phase

After the training, the model is saved and uploaded on the watchdog, making the IDS to be fully active in the network. The watchdog has to be strategically placed to have a high probability of capturing malicious traffic. Such problem has been fully discussed in [6], so we focused our attention on the creation of the datasets used for the analysis and on the IDS performance rather than its positioning. A setup based on a single watchdog has been considered, thus avoiding the replication of collected data. From the point of view of the BLE Mesh network, the watchdog behaves as a relay that mirrors the traffic received into two different flows. The first flow is left unaltered, proceeding towards the original destinations, while the latter is split into time windows, with tunable duration, and used for collecting the same feature set as in Eq (1). After a prefixed interval, all these windows are inspected by the IDS module that labels each of one them using the model created in the previous phase, being able to promptly notify a running attack.

As indicated before, this IDS requires for its training some previous knowledge of the attacks to detect, in form of datasets of BLE Mesh network traffic in presence of attacks, which is, to the best of our knowledge, totally absent in the current literature. For this reason, we conducted several experiments in order to generate these dataset simulating a standard legit network and the two different attacks taken into examination.

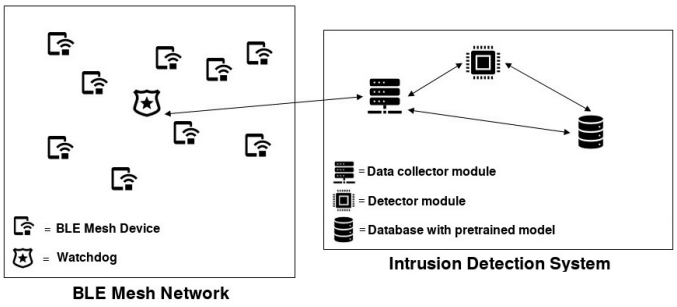

Fig. 1. Architecture of the proposed IDS.

IV. Data collection test field

To create an homogeneous test field able to reflect the features of a real-case development of a BLE mesh scenario, as

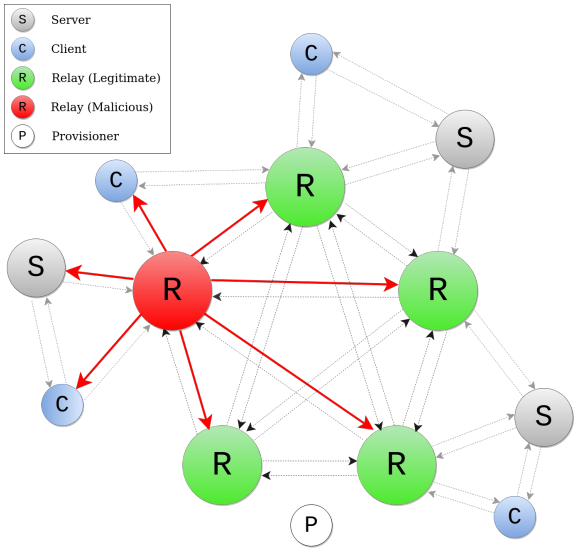

Fig. 2. Network setup for Grey hole and Black hole attacks. The same setup was built for the Legitimate simulation with a legitimate relay instead of the malicious one.

closely as possible, we decided to adopt for all our experiments the ESP32 boards from Espressif Systems [7], enabling the possibility of conducting the tests over a real IoT environment.

Moreover, the BLE Mesh standard development kit (ESP_BLE_MESH SDK) of the Espressif for ESP32 boards has been officially certified by Bluetooth SIG Alliance [8], thus allowing any BLE Mesh network built with ESP32 stack to be considered standard compliant. We decided to use the full BLE Mesh stack, adopting the Generic Level model as the application layer of the implementation. All the code produced for the experiments can be found at [9] while the code of the IDS is located at [10].

To extract the traffic data from the ESP32 boards for the creation of the datasets and for the intrusion detection analysis, we implemented a **Data Collector** module with the purpose of reading and monitoring the log output of the boards and storing all the useful data. More precisely, the Data Collector first identifies all the ESP32 boards connected via USB and for each of them launches the monitor function.

Such module has been left entirely agnostic and independent from the rest of the IDS to collect data from both the Model and Network layers, highlighting the relay nodes' work. To perform in a totally distributed fashion, the Data Collector uses a Network Time Protocol client [11] to save the offset between the NTP server and the computer, thus preserving the happened-before relation between packets.

In the next sections we discuss the various scenarios implemented to collect the data, their topology and the attacks performed on it.

A. Legitimate mesh network setup

In this setup a classical BLE Mesh network has been implemented with 13 BLE devices, of which 3 servers, 4 clients, 5 relays and a single provisioner, role that in our experiments will always be performed by a mobile phone and that will always have the address *0x0001* being the first element created by the network. To make the network fully working, the provisioner subscribe each server to a subscription group defined by a static address (*0xC001* in our experiments) and also assigns a publication address to each client (the broadcast address *0xffff* in our experiments).

Although is possible to enable the relay feature in every single role within a BLE mesh network, during these experiments only the roles defined as relay had the actual possibility to forward the message. Thus, the resulting bidirectional graph from the traffic point of view within the mesh network does not have the quality of being fully meshed, but allows the data to be taken and studied in a homogeneous way without a loss of generality.

To simulate a real usage of a BLE mesh network, each client within this experiment has been set to periodically send a GET or acknowledged SET broadcast message to all available servers on the network, according to the Generic Model Level. In the experiments, clients randomly choose the message type and the time instant in which it is sent, while the parameters of the acknowledged SET are incremental and the destination addresses of all their messages is always the broadcast address, defined by the standard as *"0xffff"*. The collection process has been performed in 6 hours long sessions, representing the possible output of a daily data-collection procedure, used to analyze the traffic or update the IDS model in a real IoT environment

B. Targeted Grey Hole attack network setup

In this experiment a particular type of attack called *Grey Hole*, carried out by a malicious node.

This scenario has been kept similar to the previous one, the legitimate case, avoiding the introduction of unwanted biases in the features' values, not strictly depending on the attack scenario under analysis. The only difference is about the introduction of a malicious relay node, impacting the messages of a particular server node chosen as a target.

This attacker decides whether or not to forward messages to be of the victim by running the Algorithm 1. Since this attack targets only one of the servers, the others elements connected to the malicious relay are not affected at all, having their messages correctly forwarded for the whole duration of the experiment. Indeed, the attack follows a binomial distribution with a success probability of 0.5, thus fulfilling the partial loss of victim's packets and so the partial denial of service .

C. Black Hole attack network setup

This final experiment implements a complete denial of service over an entire area, as also shown in Figure 2. In this case, the malicious relay impacts the communications of several devices. While some of the latter could eventually keep communicating with the rest of the network, the remaining affected devices are completely excluded from any form of service, ending up in a shadow zone

Algorithm 1: Targeted Grey Hole attack function.

Result: Packets incoming from a specific server identified by its network address are dropped with an uniform coin toss probability.

$threshold \leftarrow$ half of the maximum value obtainable by the random function;

$target\_address \leftarrow$ network address of the victim;

upon *receive_new_packet(packet)*:

  if $esp\_random() > threshold \wedge packet.src = target\_address$ **then**

    $drop\_the\_packet(packet)$;

  else

    $forward\_the\_packet(packet)$;

  end

The attack function of this experiment was run on a single relay, according to Algorithm 2. In real situations this behaviour is one of the most detectable yet it is hard to distinguish whether it is caused by a real attack or by a malfunctioning inside the network.

Algorithm 2: Black Hole attack function.

Result: After the timeout no more packets will be forwarded inside the network.

$start\_time \leftarrow get\_current\_time()$;

$timeout \leftarrow 300000$; `/* 5 minutes (used to set the node up) in ms */`

upon *receive_new_packet(packet)*:

  $start\_time \leftarrow get\_current\_time()$;

  if $start\_time > timeout$ **then**

    $drop\_the\_packet(packet)$;

  else

    $forward\_the\_packet(packet)$;

  end

After the provisioning of the network, the attack begun and the malicious relay started to drop all the incoming messages, preventing all the nodes connected to communicate with the rest of the network.

The traffic generated before the attack represents a fully legit behaviour of the network and so it was split from the rest of the attack datasets and treated as a stand alone dataset labelled as fully legit ($y_0 =$ "$Legit$"), while the rest of it was labelled as ($y_1 =$ "$BlackHoleAttack$").

V. Results and Performance Evaluation

A. Datasets analysis

Once the collection of the datasets was completed, it was possible to send them as input to the learning phase of the classifier. While the analyses have been repeated several times, each time modifying the size of the temporal observation window of the network traffic (t), we show in our results the generic time window of $t = 10$ seconds, except for the ROC curve, where different time windows of 7, 10 and 30 seconds are shown to highlight the improvement of the performances over variation of the time windows.

The **principal components analysis (PCA)** shown in Figure 3 was used in the first instance to reduce the components of the generated time windows to two single components and then each different dataset has been marked with a different color, where the blue is the legitimate dataset, the orange is the Black Hole and the green is the Grey Hole proving that despite some overlapping areas there are some distinct zones where the different datasets are easily distinguishable. This distribution in network traffic is hence highly influenced by some existing factors altering the behavior of the network, ascribable to the attacks made during the different experiments.

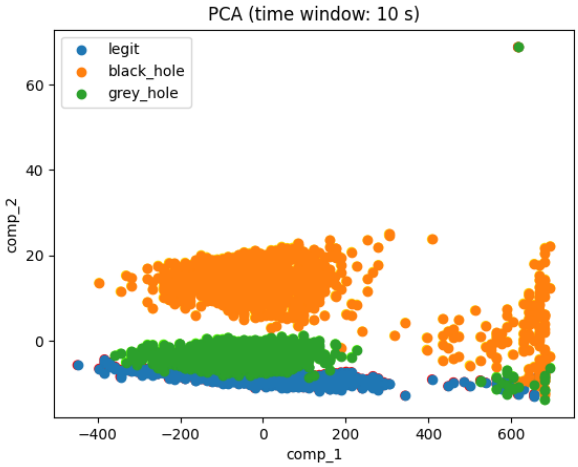

Fig. 3. Principal component analysis of the dataset ($t = 10$).

B. IDS Evaluation

We used two main tools to estimate the overall performances of the Intrusion Detection System: the **confusion matrix** and the Receiver Operating Characteristic (**ROC**) curve.

The confusion matrix in Figure 4 shows that on the first columns there are always few false negatives, i. e. there are few occurrences of Black and Grey Hole sample considered as Legit. This means that the model can make a distinction with a really good accuracy, but it has serious issue when it comes into the distinction between Grey Hole and Black Hole.

Finally, the ROC curve metric is considered to evaluate the classifier output quality. As shown in Figure 5 the results are quite coherent with the ones of the confusion matrix even in binary classification. Moreover, the IDS was able to distinguish with an high accuracy the probes evaluated in each time window considered, bringing to the conclusion that has a strong capability of detecting an attack within this network. The fact also that the performances are better as the size of the time window increases suggests that more researches toward an optimal size should be conducted.

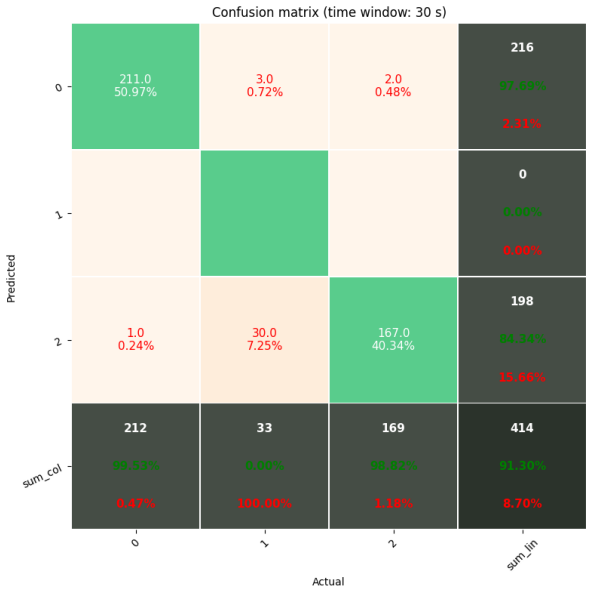

Fig. 4. Confusion matrix ($t = 30$).

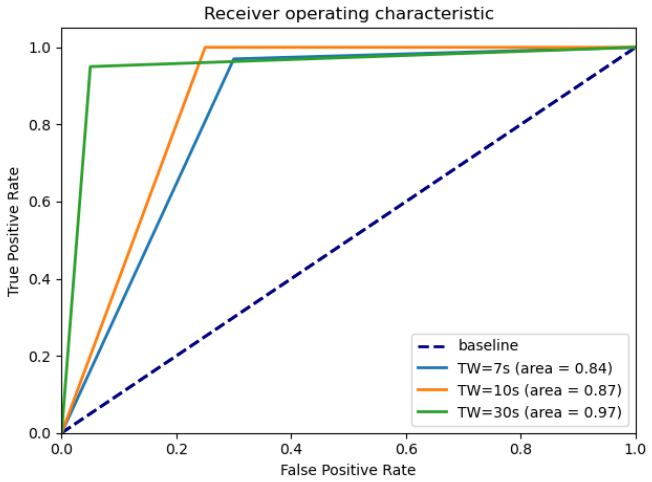

Fig. 5. Receiver Operating Characteristic comparison between different time windows.

VI. Conclusions

In this work, a new proposal for an Intrusion Detection System specifically designed for the Bluetooth Low Energy mesh networks was discussed and analyzed, thus introducing many innovative aspects in the analysis of networks.

The presented model shows that there is a series of intrinsic factors in the network traffic that allows us to understand if there are any ongoing DOS attack.

To allow the creation of valid classification patterns, several experiments have been conducted that have led to the creation of an agnostic data taking system capable of extracting many fundamental parameters to describe a BLE mesh network. The datasets generated by these experiments, especially those dedicated to attack scenarios, represent, as far as we know, the first attempt in the literature to characterize different traffic patterns in the BLE mesh network environment, being a valuable contribution for developing new tools and methodologies to correctly categorize old and new types of threats in this strongly growing IoT branch.

By analyzing the results, the IDS proves that there are actually strong correlations between the time windows. This allows the detecting of the ongoing anomalies with a strong accuracy, but its performances strongly degrades when it comes the time to understand what exactly the aforementioned anomalies are. Therefore, while the IDS has proved to have excellent potentialities and while this approach can be successful even in large implementations, the experiments done so far have been insufficient to give a more precise accuracy range.

Future works can consider to implement additional features not taken into account, as for instance the average value change of the SEQ field of packets with the same value of the source field, or to study new attacks such as the malformation of the TTL value in the packets [12], that can lead the rejection of the message.

Finally, to validate the effective ability to implement such IDS inside an IoT environment, further activities to evaluate both electrical power and of the computing capacity should be carried out.

References

[1] B. S. Alliance, "Standard bluetooth mesh specification," https://www.bluetooth.com/specifications/mesh-specifications/.

[2] L. Almon, F. Álvarez, L. Kamp, and M. Hollick, "The king is dead long live the king! towards systematic performance evaluation of heterogeneous bluetooth mesh networks in real world environments," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, Oct 2019, pp. 389–397.

[3] A. Adomnicai, J. J. A. Fournier, and L. Masson, "Hardware security threats against bluetooth mesh networks," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[4] L. K. Bysani and A. K. Turuk, "A survey on selective forwarding attack in wireless sensor networks," in *2011 International Conference on Devices and Communications (ICDeCom)*, Feb 2011, pp. 1–5.

[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[6] M. Krzysztoń and M. Marks, "Simulation of watchdog placement for cooperative anomaly detection in bluetooth mesh intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102041, 2020, modeling and Simulation of Fog Computing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X19301728

[7] E. Systems, "Home page," https://www.espressif.com/.

[8] Bluetooth.com, "Esp ble mesh certification," https://launchstudio.bluetooth.com/ListingDetails/94304.

[9] A. Lacava, "Github repository of the ble mesh network," https://github.com/BE-Mesh/esp32-ble-mesh-std.

[10] E. G. Andrea Lacava, "Github repository of the ids for ble mesh," https://github.com/BE-Mesh/BLEMeshIDS.

[11] U. D. D. Mills, "Network time protocol version 4: Protocol and algorithms specification," Internet Requests for Comments, RFC Editor, RFC 5905, 05 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5905

[12] F. Álvarez, L. Almon, A.-S. Hahn, and M. Hollick, "Toxic friends in your network: Breaking the bluetooth mesh friendship concept," in *Proceedings of the 5th ACM Workshop on Security Standardisation Research Workshop*, ser. SSR'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–12. [Online]. Available: https://doi.org/10.1145/3338500.3360334