ZBDS2023: A multi location Zigbee dataset to build innovative IoT Intrusion Detection Systems

Olivier Lourme, Gilles Grimaud, Michaël Hauspie

ZBDS2023: A multi location Zigbee dataset to build innovative IoT Intrusion Detection Systems

Olivier Lourme, Gilles Grimaud, Michaël Hauspie
Université de Lille, CNRS, IRCICA, Centrale Lille, UMR9189 - CRIStAL
Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille
firstname.lastname@univ-lille.fr

***Abstract*—The emergence of the Internet of Things (IoT) model featuring different types of wireless networks made of many different constrained devices conducts researchers to face new security challenges to protect these networks. Indeed, they need to build and evaluate dedicated monitoring tools and Intrusion Detection Systems (IDSs). To succeed in these goals, quality datasets including benign and under attack situations are necessary for all IoT protocols. However, if we consider for instance the smart-home sub-market dominated by Wi-Fi, Zigbee and BLE, only complete Wi-Fi datasets can easily be found today. To specifically overcome the lack of Zigbee datasets and contribute to the development of related IDSs, this paper presents ZBDS2023, a 10-day realistic Zigbee dataset made publicly available. Fully documented with metadata, it has been collected from a real populated smart home equipped with 10 recent Zigbee lighting devices. Some periods of capture are free of attack, allowing to build normality models, and some include various labelled attacks, enabling the evaluation of different intrusion detection strategies. Also, as an original second contribution, each emitted frame is captured by one to four demodulating passive probes distributed in the house. Besides providing redundancy concerning MAC layer data, the corresponding values of Received Signal Strength Indicator (RSSI) have also been made available. Being a physical feature, RSSI cannot be easily impersonated and as such, it is a priori a good candidate for participating in fingerprints feeding spoofing detection systems. Moreover, its extraction is uncostly and available in many wireless technologies. However, exploitation of RSSI time series is not trivial, especially in populated buildings. A third contribution using our dataset evaluates a naive attack detection system to serve as a baseline for future works.**

***Index Terms*—IoT security, Zigbee, dataset, Intrusion Detection Systems, spoofing attacks, RSSI**

I. Introduction

A. Context

Appeared with the 21st century, the Internet of Things (IoT) introduced a new model in data processing by ensuring a bridge between the physical world and the virtual one of supervision. Used actuators and sensors are built upon MCUs featuring modest processing and wireless communication resources so that they can achieve their well-defined task and be organised in networks of dedicated topologies and technologies. Most of the time, these networks are connected to the Internet directly or via gateways to ensure their supervision by different means (website, cloud platform, smartphone application). Fostering better comfort and fast decision making, IoT pervades not surprisingly almost all sci-tech fields rendering them "smart": smart agriculture, smart city and smart home for instance [1].

Along with this huge deployment, security has often been neglected. The famous CIA triad "Confidentiality, Integrity, Availability" is certainly considered in IoT but to variable degrees. Indeed, the heterogeneity and specificity of protocols and topologies, the variety of deployed constrained devices, as well as commercial pressures on manufacturers lead to solutions that are less mature and less standardized than those available in traditional Information Technologies (ITs). This exposes a new and often targeted attack surface, potentially leading to private data leaks, device takeovers and denies of service [2]. At last, in the IoT smart-home subdomain, the appropriation of devices by users with no security culture, looking for new functionalities deployed with low friction, is an aggravating factor. So, even if all stakeholders in the IoT ecosystem are more and more concerned with designing, selling and using safe devices, there is a need in IoT for Intrusion Detection Systems (IDSs) as a first line of defence.

B. Motivation

IDSs for IoT are an active research field in the cybersecurity literature [3]. A realistic dataset shared and recognized as a reference by researchers is a mandatory tool, first to design and assess the IDS itself and then to compare it with existing solutions. Trying to build an original IDS dedicated to Zigbee, we faced the following paradox: although Zigbee, along with the other Wireless Personal Area Networks (WPANs), like BLE and Z-Wave, feature approximately the same number of active connections than Wi-Fi devices[1], it was impossible to find a decent Zigbee dataset. By opposition, more and more interesting Wi-Fi datasets are becoming available in the literature. It seems there is a bias that makes IT researchers choose Wi-Fi for their IoT security studies because they are already comfortable with it and its TCP/IP environment. The contributions of this paper are threefold:

- to design and assess Zigbee IDS in a reproducible research spirit, we propose ZBDS2023, a 10-day realistic Zigbee dataset featuring complete exchanges between 10 Zigbee lighting devices deployed into a populated smart home, featuring benign and under attack situations. The dataset is provided with metadata detailing its elaboration and labels concerning the injected attacks,

[1] https://iot-analytics.com/number-connected-iot-devices/

- each IEEE 802.15.4 frame emitted by a device is captured by one to four demodulating probes in the house (depending on device coverage in a given environment), providing some redundancy concerning MAC layer data (source identifier, frame type and length, etc.). Moreover, each device frame emission is also characterised by a set of the corresponding values of Received Signal Strength Indicators (RSSIs), measured by the probes. RSSI being a location-dependant hence a physical feature, fingerprints based on one or several of them are hard to imitate by an attacker. This multiple RSSIs availability provides degrees of freedom for study and makes our dataset an original tool to assess innovative spoofing detection solutions,
- RSSI is a convenient, easy to extract physical feature. It is also present in most wireless technology, giving hope for cross-technology intrusion detectors. However, it varies a lot inside buildings, especially if they are populated. Illustrating that point and making a first usage of the dataset, a simple RSSI-based IDS is proposed to serve as a baseline that future works could refer to.

C. Paper organisation

The remainder of the paper is organised as follows: in Section II, we present the fundamentals of Zigbee permitting to understand the rest of the paper. Identically, Section III provides insights regarding RSSI. Section IV comments the architecture of our testbed and the data collection process. In Section V, the injected attacks are documented. Section VI presents a naive spoofing detection using our dataset. Works related to Zigbee datasets are exposed in Section VII. At last, Section VIII concludes this paper.

II. Zigbee and ZLL profile

A. Presentation

Appeared in 2005, Zigbee is a standard for IoT developed by the Connectivity Standard Alliance (Samsung, Philips, Texas Instruments, Ikea, etc.). It aims at providing a two-way, reliable wireless communication protocol for devices within low-power and low-cost WPANs, wherein the range is typically under 100 meters. Most of the time, it operates in the free licensing 2.4 GHz ISM band over one of the sixteen 2 MHz-wide non overlapping channels; the modulation is O-QPSK and the throughput is up to 250 kbps. To ensure interoperability between devices issued from different manufacturers, some public application profiles have been introduced, depending on the use case Zigbee is intended for, e.g., Home Automation or Industrial Plant Monitoring. The testbed from which our dataset is extracted works with the Zigbee Light Link profile (ZLL), dedicated to consumer lighting solutions: mood lighting, energy monitoring and light management via occupancy sensors are example of ZLL functionalities. ZLL is part of "Zigbee Pro", the Zigbee version dating 2007. A "Zigbee 3.0" version appeared in 2015. Backward compatible, it aims at providing safer device associations and unifying more Zigbee products. However, we were able to check that the 10 Philips Hue lighting devices we purchased between 2020 and 2022 to build our dataset were still running the 2007 "Zigbee Pro" version. Compared to other profiles like Home Automation, ZLL is a simplified one. For instance, concerning security, a ZLL network has neither "Trust Center" nor "Coordinator". The replacing "Control Bridge", that is also an Internet bridge, provides only a basic security key management, detailed later. Also, to provide a simpler device installation experience, "Touchlink commissioning" has been introduced in addition to classical IEEE 802.15.4 association.

B. Zigbee stack

The Zigbee stack is constituted of 4 layers as indicated on Figure 1: Lower ones are Physical (PHY) and Medium Access Control (MAC) layers. Upper ones are Network and Application layers.

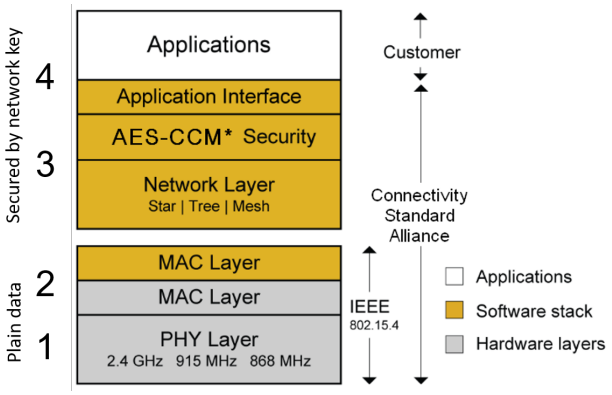

Figure 1. Zigbee stack [4]

1) Lower layers: Zigbee Pro relies on IEEE 802.15.4-2003 for the two lower layers of the stack. Security options of this IEEE standard are not used with Zigbee, allowing an attacker to get a lot of interesting information by a passive eavesdropping on the used channel: number of devices and their identifiers, length of their frames, inter-frame time interval, etc. There is no authentication planned at these lower layers, which means an attacker can spoof a legitimate node, forging repeated IEEE 802.15.4 frames with its source identifier to flood a victim with inappropriate messages, causing to it and to the network resource depletion.

2) Upper layers: the Zigbee standard defines the two upper layers of the Zigbee stack. Relying on a modified AODV protocol, the network layer allows self-organising networks, in star, tree and mesh topologies. This latter, illustrated in Figure 2, permits range extension and is self-healing. Routers are plugged on mains, they are typically light bulbs, smart plugs or the Internet Bridge. End devices like remotes or motion sensors often run on batteries planned to last about 10 years.

C. Security

1) Description: security is introduced at Network layer. With ZLL, to emit and receive Zigbee messages like routing data or on/off commands, devices must possess an AES 128-bit credential "network key" shared with all devices in the

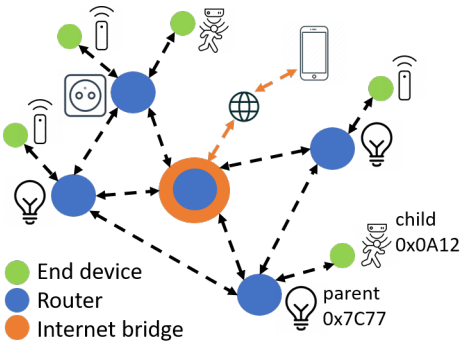

Figure 2. A Zigbee Light Link network with a mesh topology

network. As depicted in Figure 3, on the transmitter side, data confidentiality is ensured by its cyphering with the network key and a "Nonce", through an AES-CCM* block, that also computes a "Message Integrity Code" (MIC), checked at arrival. The Nonce features a 32-bit "Frame Counter" that is incremented every time a frame is transmitted, constituting an anti-replay protection. However, rigorous implementation of all these protection mechanisms is only guaranteed with high-end Zigbee devices and when low current consumption is not a requirement [5].

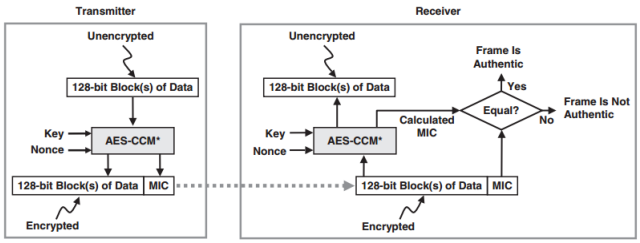

Figure 3. Data integrity and data confidentiality in a Zigbee exchange [6]

2) MAC commissioning vulnerability: during the commissioning phase, a device joins the network. We focus here only on commissioning permitted by the still widespread IEEE 802.15.4 MAC association procedure. To get associated, a device issues Beacon Request frames on all channels. The first router in the vicinity to respond with a Beacon frame will handle the association process. At this occasion, the device gets a 16-bit logical identifier as well as the aforementioned network key, ciphered by a symmetric "Light Link Commissioning key" preinstalled on all ZLL devices in the world and available on the web with a simple search. Passive listening of the device association process (with Wireshark informed about the commissioning key) provides easily the network key. This is still used today to respect backward compatibility but it constitutes a major security vulnerability because as soon as an attacker gets this key, he has access to all messages or may compromise the whole network [7]. Forcing a user to rerun a device association can be achieved by jamming the device or flooding it.

So, given the security context we exposed, it is crucial to trust a device with more than an easy-to-forge identifier or the effective possession of an easily obtainable key. Using features extracted from the PHY layer will help building a hard-to-mimic device fingerprint that can be used in an IDS trying to defeat spoofing attacks. Our dataset offers this option by capturing the frames from several locations providing a set of several RSSIs per device for each frame it emits.

III. RSSI BACKGROUND

RSSI is a measure of the mean power arriving on a receiver, for instance an IDS probe, converted to dBm [6]. In a naive open free space approach, at a fixed frequency, the received power is proportional to the emitting power and to the inverse of the square of the emitter-receiver distance. If emitter and receiver stay at their dedicated location and the emitter does not vary its emitting power, the receiver measures an almost constant RSSI value. Associating this latter with the emitter logical identifier, we get an entry for a table that can be further used for authentication purpose defeating intrusion attempts. Contrary to features issued from MAC layer, RSSI is a physical one and as such, it is difficult (but not impossible) for an attacker to mimic the one of a legitimate node. A frame with a certain identifier but with a non-related RSSI is symptomatic of a masquerade attack. On the contrary, many identifiers associated to a unique RSSI are symptomatic of a sybil attack. RSSI has the advantage of taking the form of a unique scalar, providing a costless and immediate access, not needing any preliminary and heavy device characterization [8], unaffordable in smart-home contexts. Also, RSSI is available in many wireless technologies needing Clear Channel Assessment and signal quality estimation, giving hope for generalizing a RSSI-based IDS to other wireless technologies. Typical RSSI parameters of a specific probe are given in Section IV-C.

This approach remains valid inside static buildings, even if obstacles create signal reflection, absorption and diffraction, inducing multipath fading. To introduce robustness and compensate for nodes (devices or attacker) having too close RSSI values and thus preventing efficient authentication, one can consider a distribution of several probes to characterize each emitter by a tuple of RSSIs, an approach deeply explored in [9]. Sadly, the environment test of this work is necessarily static and as such not representative of a populated smart home where inabitants, doors and furniture move frequently, constantly redrawing the environment. As shown in Figure 5 that depicts the RSSI associated with 0x0A12 identifier, captured from a single probe on a whole day, environment changes cause across time great RSSI volatility, preventing to easily associate an RSSI value to a device identifier (RSSI is steady only when the house is asleep). Besides, in case of spoofing attacks, additional RSSI values due to the attacker are merged with the volatile RSSI values of the legitimate node, making difficult to guess what is responsible for what in the global RSSI profile. However, with the help of the proposed dataset, the security community will be able to design some IDSs with appropriate algorithms that circumvent these difficulties.

IV. Dataset and testbed

A. Dataset characteristics

Guided by the criteria of datasets exposed in [10] and [11], we then designed a Zigbee dataset presenting the following characteristics and qualities:

1) ***length****:* 10 days; the capture started June 30, 2022 at 17:00 (CEST) and ended July 11, 2022 around 09:00 (CEST);

2) ***place of capture****:* to build a realistic smart-home Zigbee dataset, the capture occurred in a 100 m^2 two-storey house where people were normally evolving and where doors were regularly opened and closed. With walls, it formed an always changing set of obstacles. Also, a microwave oven was used in the kitchen and Wi-Fi and BLE networks were coexisting with the main Zigbee experience. The devices were left at their dedicated location emitting with constant power, the probes used for captures were not moved either;

3) ***devices****:* 10 Philips Hue lighting devices (dating 2020 to 2022) were used; 4 types of devices among 6 are represented each by two instances;

4) ***captures****:* thanks to 4 passive probes distributed in the house, the frames are captured up to 4 times, giving beyond classical MAC layer data up to 4 RSSI information; the dataset is organised in mergeable elementary PCAP files sorted per probe and hour; with approximately 1000 frames per minute, it is about 2.8 GB uncompressed;

5) ***accurately labelled attacks****:* the traffic is mostly benign allowing to build a traffic background profile but 50 attacks documented in Section V have been injected to assess IDS strategies;

6) ***security assessment****:* on July 9, 2022 at 15:39 (CEST), a second dimmer switch (DS2, see Table I) was associated to the Zigbee network. The whole association is captured in the dataset. Knowing the Light Link Commissioning key and using Wireshark, the network key is easily obtained, giving access to deciphered messages of upper layers (see Section II-C2); also, with that vulnerability knowledge, one may now consider building a malicious device that can get authenticated in real ZLL networks;

7) ***metadata****:* the whole dataset constitution and use are documented in this paper;

8) ***availability****:* the ZBDS2023 dataset is publicly available along with a Gitlab "Resources Repository" to get started [12].

B. Testbed

Figure 4 shows the location of the 4 probes and 10 devices used in the testbed enabling the dataset. The following sections detail both types of resources.

C. Probes

Captures are made from 4 different locations in the house thanks to 4 probes referenced in Figure 4 as RPI1, RPI2, RPI3 and RPI4. Each one is built upon a low-cost CC2531 USB dongle[2] providing demodulated data, connected to a Raspberry Pi. The Texas Instruments firmware `fw_cc2531.hex` we flashed to the four dongles replaces the two Frame Check Sequence (FCS) bytes of each IEEE 802.15.4 frame by a single bit of FCS correctness and an 8-bit signed RSSI value. A -73 dB offset must be added to this latter to get the actual RSSI value[3]. To benefit from this FCS replacement and hence accessing to RSSI values with tools relying on Wireshark (Pyshark library, etc.), one must tell Wireshark in its IEEE 802.15.4 preferences that this Texas Instruments specificity is set, otherwise all FCSs will be indicated as bad.

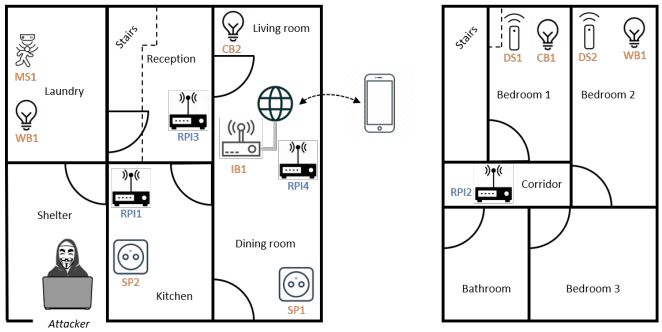

Figure 4. Location of probes and devices in the house of test: ground floor (left) and first floor (right)

On the software side, ccsniffpiper[4] is a tool compatible with the aforementioned firmware, allowing to trig captures and obtain PCAP files. We automated by a bash file the process of starting new captures every hour and naming them accordingly. Once unzipped, the dataset is made of 4 folders named `rpi1`, `rpi2`, `rpi3` and `rpi4` containing the one-hour captures made respectively by the probes RPI1, RPI2, RPI3 and RPI4. Each elementary capture is a PCAP file named like this one:

`rpi2-2022-07-01-18-00-00.pcap`

This name means that the capture has been made from RPI2 probe and that it started at 18:00:00, on the 1st of July 2022 (CEST). All PCAP files cover one hour of capture. They may be merged together, for instance with Wireshark, to obtain captures of the desired duration.

To date, there is no tool to regroup the 4 frames of each emission but building it should not be a complex task: Before starting the capture, the four probes were synchronized using NTP and the 4 different epoch times of arrival are available in the PCAP files (the magnitude order of times of arrival differences is 10^{-4} s). Moreover, most of the time monitoring systems compute statistics on sliding windows of several seconds and correlations start only from those aggregated values. Also, determining the best located probe among the 4 in terms of frame capture efficiency should be considered to design a practical and low-cost single probe IDS.

D. Devices

1) Description: characteristics of devices exposed in Figure 4 are given in Table I. The IEEE 802.15.4 short 16-bit

[2]RSSI sensitivity=-97dBm, range=100dB, accuracy=±4dB, step size=1dB

[3]https://www.ti.com/lit/ug/swru191f/swru191f.pdf, pages 230 and 233

[4]https://github.com/andrewdodd/ccsniffpiper

identifier is the one obtained after an association procedure. All devices already had such an identifier when the capture session started, except DS2 that got associated later to capture its association process. Some devices are Reduced Function Devices (RFDs), they can only be end-devices in the Zigbee network (see Figure 2). They run on battery, so they sleep most of the time but wake up on specific events, making an IEEE 802.15.4 Data Request frame to their parent for getting data arrived at their intention during their sleep. The other devices are Full Function Devices (FFDs), they are powered from mains and have router capabilities.

For all devices, PAN ID is `0xB7C5` and Extended PAN ID is `47:4D:CE:61:BF:05:E6:EF`. The used channel number is 20.

2) Bindings: The following binding relationships (that may be different from parent-child ones) were set with the Philips Hue smartphone companion application. These bindings provide interactions between devices, participating to a realistic dataset:

- WB1 lights up when MS1 detects a movement,
- CB2 is voice-controlled by a Google Home thanks to IB1,
- DS1 controls CB1,
- DS2 controls WB2 (after DS2 joined the network).

Besides, the Philips Hue companion application for smartphone was widely used to control all devices, from inside and outside the house, thanks to IB1.

V. Injected attacks, threat model

To foster the development of Zigbee-dedicated IDSs and also spoofing detection systems, the proposed dataset includes different types of attacks. These ones have been chosen as representative of what an attacker may undertake with little expertise and resources. Most attack shapes are based on message flooding. The resulting resource depletion and unresponsiveness shown by devices and network may be a prejudice sufficient by itself or it may be a first step in a sequence of attacks like the one leading to network key disclosure, detailed in Section II-C2, or like others described in [13]. Since we used high-end Zigbee devices, not all attacks ended up being successful but their detection is nevertheless a crucial point and this is what our dataset is intended for. The following conditions were observed:

- like devices and probes, the "neighbor" attacker remains at his fixed position, indicated in Figure 4. He is able to take several identities by forging malicious packets that are injected at constant emitting power using Killerbee offensive framework[5] on an Ubuntu laptop, the transceiver being a fifth CC2531 USB dongle, flashed this time with Bumblebee offensive firmware[6],
- the attacker has no knowledge of the network key. He can only eavesdrop the network to gain knowledge on it or inject some IEEE 802.15.4 messages to perform for instance flood-based attacks or replay ones.

[5]https://github.com/riverloopsec/killerbee

[6]https://github.com/virtualabs/cc2531-killerbee-fw

Frames are captured from several locations. First, it brings an appreciable redundancy when designing monitoring systems exploiting MAC layer data like in [14] and secondly, one RSSI or a correlation of several of these PHY layer features should be an interesting starting point to build spoofing detection systems.

During the 10 days of capture, we initiated on 6 of them 10 sessions of 5 attack types, each session being numbered between 1 and 10 and each attack type being labelled from A to E. Performing several times the same attacks is justified because the house environment is constantly changing. Table II gives the description of attack types.

Attacks are characterized by their session, their type and their date of start. In Python, using Pandas library, the following dictionary defines for example the E-type attack that occurred during the 9th session:

```
{
  'sess': '09',
  'type': 'E',
  'start': pandas.Timestamp('2022-07-08 20:25:10+02:00')
}
```

The complete list of attacks presented that way is provided in the file `attacks_references.py` in the "Resources Repository", accessible from [12].

VI. Use case

To illustrate the use of the dataset, we propose in this section a naive RSSI threshold-based IDS aiming at detecting spoofing attacks. On the day of July 8, 2022, the attacks 8C, 9C, 10C, 8E, 9E, 10E (cf. Table II) were injected into the network besides its normal activities. During these latter, the attacker used the identifier of the legitimate child (MS1) to issue Data Requests to WB1, the parent of MS1. Both types of attacks last one minute but the "C" ones have a low rate (12/min) whereas the "E" ones have a high rate (500/min). To proceed, we made a 24-hour PCAP file from data captured by only the RPI2 probe. First, we pre-processed the file, excluding malformed frames, ACK frames, frames with bad FCS and frames with outliers concerning RSSI. Then, we just kept the frames associated with the 16-bit identifier of MS1, i.e., 0x0A12. To set up a basis for the reference RSSI value of MS1, we used the relative stability presented by the RSSI during night hours where the environment is stable. So, we computed the mean of the RSSI during 12PM and 6AM (-86 dBm).

As depicted in Figure 5, instant RSSI is volatile and appears inappropriate for simple models based on thresholds. To circumvent this and also to take into account the temporal dimension of RSSI series, we computed the RSSI moving average using sliding windows of 30 values, a number sufficient to smooth RSSI but small enough to remain sensitive to abrupt changes. The resulting plot is given by Figure 5. Then we had to choose a pair of thresholds around the previously computed RSSI night mean; concerning attack detection, if the RSSI moving average goes out of the band limited by the thresholds, we consider there is an attack and we raise an alarm. The width of the band was chosen manually to ±4 dBm as it gives the best detection metrics. This is a non-realistic approach but the

Table I
LIST OF DEVICES PRESENT IN THE ZIGBEE DATASET

Ref.	Type	16-bit id.	MAC address	RFD/FFD
MS1	Motion Sensor	0x0A12	00:17:88:01:0B:CD:0C:32	RFD
SP2	Smart Plug	0x0C06	00:17:88:01:08:D8:9B:69	FFD
IB1	Internet Bridge	0x22FD	00:17:88:01:05:11:BD:4A	FFD
DS1	Dimmer Switch	0x46EA	00:17:88:01:08:F0:4C:D7	RFD
CB2	Color Bulb	0x5EBA	00:17:88:01:06:A1:66:0B	FFD
WB2	White Bulb	0x7B9E	00:17:88:01:08:B6:20:FE	FFD
WB1	White Bulb	0x7C77	00:17:88:01:08:BF:D3:B7	FFD
DS2	Dimmer Switch	0x88C2	00:17:88:01:0B:DA:40:CE	RFD
SP1	Smart Plug	0xA2AB	00:17:88:01:08:D8:B9:B8	FFD
CB1	Color Bulb	0xA8DF	00:17:88:01:06:A1:5A:00	FFD

Table II
TYPES OF CONDUCTED ATTACKS AND THEIR DESCRIPTION

Type	Attack shape	Attack details	Comments
A and B	Assoc. Request Flood	By emitting a Beacon request frame, the attacker asks all routers to inform him about the characteristics of the network they belong to. The one responding first, generally the Internet Bridge, will handle the association process. At this occasion, the attacker adopts a random 64 bits MAC address and gets eventually a 16-bit identifier. A flood of such actions is a sybil spoofing attack consuming the available pools of RFD and FFD 16-bit identifiers.	Duration $\approx$ 30 s. Between 35 and 50 Association Requests. A Permit Join message of duration 30 s was previously sent to all routers by the smartphone companion application.
C	Replay Attack	In this attack, the attacker masquerades the Internet Bridge IB1 to achieve a device takeover that may be successful in case of degraded implementation of security mechanisms (no frame counter). An elementary PCAP file captured during 10 s several orders from IB1 to power on and off CB2, among other normal exchanges. This elementary file is replayed 6 times in a row by the attacker. In the 10-second elementary PCAP file, frames having other purpose than powering on and off CB2 were captured too. For instance, in this elementary file, MS1 performs 2 normal Data Request to WB1, so that makes 12 Data Request issued from the attacker during the attack. This side effect induces a kind of low-rate type "E" attack.	Duration $\approx$ 1 min. The power-on and power-off orders were issued from the smartphone companion application.
D	Data Request Flood	In this masquerade attack, the attacker takes the identifier of DS1 to forge hundreds of Data Requests frames targeting CB1. This latter is not the parent of DS1, so it will issue "Leave" frames to DS1. If DS1 is configured to respond to "Leave" orders, it is a mean for the attacker to make it leave the network.	Duration $\approx$ 1 min. About 500 Data Requests.
E	Data Request Flood	In this masquerade attack, the attacker takes the identifier of MS1 to forge hundreds of Data Requests frames targeting WB1, the legitimate parent of MS1.	Duration $\approx$ 1 min. About 500 Data Requests.

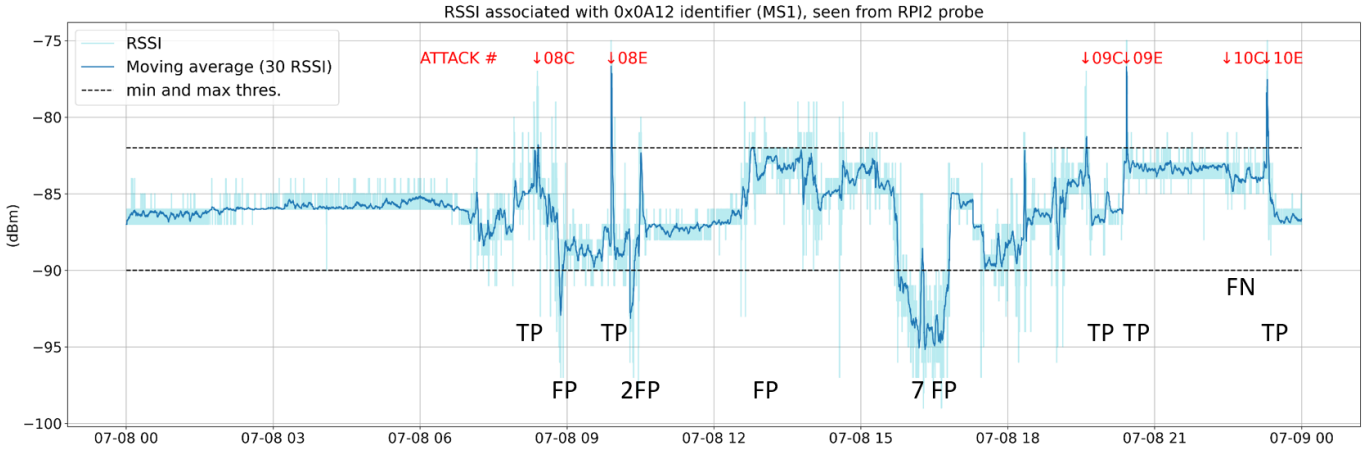

Figure 5. Test of July 8, 2022: Instant RSSI and RSSI moving average related to 0x0A12 identifier, seen from RPI2 probe

goal here is to establish a baseline that more subtle algorithms will compare to.

To establish the detection metrics, we sliced the 24 hours in 10-minute periods to correlate in each of them effective attacks and raised alarms (several alarms in a period occur for just one), establishing the numbers of True Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP) and the classical metrics Accuracy (Acc), Precision (Prec), Recall (Rec), True Negative Rate (TNR) and False Positive Rate (FPR) given by the following formulas:

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

$$Prec = \frac{TP}{TP+FP} \quad (2)$$

$$Rec = \frac{TP}{TP+FN} \quad (3)$$

$$TNR = \frac{TN}{TN+FP} \quad (4)$$

$$FPR = \frac{FP}{TN+FP} \quad (5)$$

Displayed in Table III, the results are quite interesting despite the simplicity and cheapness of the detector. We detected five attacks out of six performed. Of course, there is room for improvement concerning the False Positives but our dataset revealed that using RSSI for spoofing detection is a promising approach. Metrics should be improved with no doubt using more sophisticated algorithms that take into account the temporal essence of RSSI, e.g., Recursive Neural Networks, and that possibly integrates more features as well.

Table III
DETECTION METRICS OF THE USE CASE

TN	TP	FN	FP	Acc	Prec	Rec	TNR	FPR
127	5	1	11	91.7%	31.2%	83.3%	92.0%	8.0%

VII. EXISTING ZIGBEE DATASETS

Zigbee datasets featuring benign and attacking phases are not represented in the literature in proportion of this standard diffusion. As a consequence, many papers with "IoT" in their title only consider Wi-Fi devices, missing the IoT heterogeneity. For example, the authors of [15] propose a popular fully annotated "IoT" network traffic dataset including attacks and permitting the fine security evaluation of 45 devices. Sadly, they are all Wi-Fi-based. Likewise, the authors of [16] released a consequent dataset to evaluate how "IoT" devices expose private data to the Internet but here again the authors chose exclusively Wi-Fi devices.

On the contrary, some works like [14] and [17] considered IoT heterogeneity, manipulating big amounts of Wi-Fi, BLE and Zigbee traffic in order to build monitoring systems and IDSs. But unfortunately, their datasets have not been publicly released preventing their experiments to be reproduced.

In [18], the authors released an interesting IoT dataset caring about more than Wi-Fi as they implemented a testbed with 60 Wi-Fi, Z-Wave and Zigbee devices. Moreover, they systematically collected data from each device in several stages, for instance powering, associating and interacting as far as Zigbee is concerned. Hence, it tends to reproduce a collection of the different elementary behaviors occurring in a populated smart home. Also, they captured several IEEE 802.15.4 associations that allowed us to recover some of their Zigbee network keys. However, active attacks were again only conducted on Wi-Fi devices preventing to use their dataset for building a Zigbee monitoring tool or an IDS.

Another ambitious project, is described in [19]. Featuring raw IQ and demodulated BLE and Zigbee traffic, it is oriented toward fingerprinting and localization but makes no mention of conducted attacks.

Dataset generation with use of deep learning [20] or attack injection on a still background of PCAP files [10] could compensate for the absence of datasets or their non-practical implementation in case of large networks. These are active

topics in IT security and the possibility of their adaptation to IoT dedicated standards should be investigated without delay.

Confronted with this lack of complete Zigbee datasets, we decided to build our own, respecting quality criteria exposed in Section IV-A. We also had in mind to make it feature several times two instances of the same <*brand*, *device type*> pair[7] to foster identification not only at a device type level but also at a device instance level. Many experiments in the literature often settle just one item per device type, a non-realistic situation in case of lights for example, and as soon as the device type is detected, the authors announce it for a device instance identification. Proposing for each device some additional physical data, our dataset enables works aiming at differentiating device instances one from another [8]. At last, filling the lack of complete Zigbee datasets by creating ZBDS2023 allowed us to present a baseline IDS that will foster future works in the field of attack detection.

VIII. Conclusion and future works

Benefiting from complete and accurate datasets is important for the security community to assess new ideas in the intrusion detection topic and to compare results with each other. In this paper, we presented ZBDS2023, a public and realistic dataset filling the lack of Zigbee datasets in the literature, featuring large benign periods but also various documented attacks. Offering access to MAC layer data of each emitted frame, it may also deal with spoofing detection as it makes RSSI data available from four distributed probes, allowing to establish physical fingerprints of all devices in the presented testbed. However, if RSSI is easily accessible in many IoT wireless technologies, one of its intrinsic characteristics is its dependency to relative positions and power of emitters. To free from it, we designed a Zigbee testbed with static devices (emitting at constant power) and static probes, a choice remaining realistic in Zigbee smart-home contexts. That said, we made the attacker static and with constant emitting power as well, an option that can be discussed as basic attacker strategies to evade detection is to move and vary emitting power. That should be taken into account for a future version of the dataset. At last, in a simple but promising IDS study that can serve as a baseline for future works, we exposed the complexity of differentiating RSSI variations due to attacks and those due to the "natural" RSSI volatility that occurs in populated buildings. We make the wish our dataset will be helpful for the community to develop relevant detection algorithms facing the security challenges we exposed.

References

[1] ENISA, "Baseline Security Recommendations for IoT," 2017, library Catalog: www.enisa.europa.eu. [Online]. Available: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot

[2] A. P. O. . a. . P. Newsweek, "We're surrounded by billions of internet-connected devices. Can we trust them?" Oct. 2019, section: Tech & Science. [Online]. Available: https://www.newsweek.com/2019/11/01/trust-internet-things-hacks-vulnerabilities-1467540.html

[3] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300802

[4] SILABS, "UG103.2: Zigbee Fundamentals," p. 31, 2021. [Online]. Available: https://www.silabs.com/documents/public/user-guides/ug103-02-fundamentals-zigbee.pdf

[5] B. Stelte and G. D. Rodosek, "Thwarting attacks on ZigBee - Removal of the KillerBee stinger," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. Zurich, Switzerland: IEEE, Oct. 2013, pp. 219–226. [Online]. Available: http://ieeexplore.ieee.org/document/6727840/

[6] S. Farahani, *ZigBee Wireless Networks and Transceivers*, newnes ed., 2008.

[7] T. Zillner, "ZigBee exploited - The good, the bad and the ugly," *Magdeburger Journal zur Sicherheitsforschung*, no. 12, pp. 699–704, 2016.

[8] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device Fingerprinting in Wireless Networks: Challenges and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[9] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proceedings of the 5th ACM workshop on Wireless security*, ser. WiSe '06. New York, NY, USA: Association for Computing Machinery, Sep. 2006, pp. 43–52. [Online]. Available: http://doi.org/10.1145/1161289.1161298

[10] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On Generating Network Traffic Datasets with Synthetic Attacks for Intrusion Detection," *ACM Transactions on Privacy and Security*, vol. 24, no. 2, pp. 8:1–8:39, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3424155

[11] M. R. Shahid, G. Blanc, H. Jmila, Z. Zhang, and H. Debar, "Generative Deep Learning for Internet of Things Network Traffic Generation," in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, Dec. 2020, pp. 70–79, iSSN: 2473-3105.

[12] O. Lourme and M. Hauspie, "ZBDS2023 Zigbee dataset," Mar. 2023. [Online]. Available: https://doi.org/10.57745/NDW74U

[13] F. Sadikin, T. v. Deursen, and S. Kumar, "A ZigBee Intrusion Detection System for IoT using Secure and Efficient Data Collection," *Internet of Things*, vol. 12, p. 100306, Dec. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520301384

[14] P. Anantharaman, L. Song, I. Agadakos, G. Ciocarlie, B. Copos, U. Lindqvist, and M. E. Locasto, "IoTHound: environment-agnostic device identification and monitoring," in *Proceedings of the 10th International Conference on the Internet of Things*, ser. IoT '20. Malmö, Sweden: Association for Computing Machinery, Oct. 2020, pp. 1–9. [Online]. Available: http://doi.org/10.1145/3410992.3410993

[15] O. Alrawi, F. Monrose, A. Faulkenberry, and M. Antonakakis, "YourThings: A Comprehensive Annotated Dataset of Network Traffic from Deployed Home-based IoT Devices," 2022.

[16] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach," in *Proceedings of the Internet Measurement Conference*. Amsterdam Netherlands: ACM, Oct. 2019, pp. 267–279. [Online]. Available: https://dl.acm.org/doi/10.1145/3355369.3355577

[17] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and S. Uluagac, "Peek-a-boo: i see your smart home activities, even encrypted!" in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. Linz Austria: ACM, Jul. 2020, pp. 207–218. [Online]. Available: https://dl.acm.org/doi/10.1145/3395351.3399421

[18] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the Development of a Realistic Multidimensional IoT Profiling Dataset," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, Aug. 2022, pp. 1–11.

[19] A. Duque, M. Finet, T. Vial, and M. Humbert, "SDR4IoT BLE & Zigbee RF dataset," Mar. 2021. [Online]. Available: https://zenodo.org/record/4639390

[20] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116.

[7] For instance: <Philips Hue, A60 E27 800 white bulb>