Remotely Exploiting AT Command Attacks on ZigBee Networks

Ivan Vaccari, Enrico Cambiaso, Maurizio Aiello

National Research Council (CNR), IEIIT Institute, via De Marini, 6 - 16149 - Genova, Italy;
{name.surname}@ieiit.cnr.it
Correspondence: enrico.cambiaso@ieiit.cnr.it; Tel.: +39 010 6475 226

October 9, 2017

Abstract

Internet of Things networks represent an emerging phenomenon bringing connectivity to common sensors. Due to the limited capabilities and to the sensitive nature of the devices, security assumes a crucial and primary role. In this paper, we report an innovative and extremely dangerous threat targeting IoT networks. The attack is based on Remote AT Commands exploitation, providing a malicious user the possibility to reconfigure or disconnect IoT sensors from the network. We present the proposed attack and evaluate its efficiency by executing tests on a real IoT network. Results demonstrate how the threat can be successfully executed and how it is able to focus on the targeted nodes, without affecting other nodes of the network.

1 Introduction

The Internet is today adopted for a wide range of different purposes and by several kinds of entities, ranging from banking and stock market sectors adoption to personal use for social networking and web surfing. The Internet is indeed today populated by billions of devices of different nature. In the last years, we have seen the appearance of several categories of always connected devices: smartphones, tablets, smartwatches and healthcare devices are today only a few kinds of components of the global network. We are today experiencing a new emerging trend related to the evolution of common "analog" sensors, making them connect one each other, creating a "parallel" network based on machine-to-machine communications.

In this context, the term Internet of Things (IoT) represents a general concept relative to the ability of common sensors to collect data from the real world, hence share the retrieved information across a network, by communicating with other connected devices. IoT networks are today deployed for different purposes. The most known and adopted ones are the home automation/domotics and the industrial (Industry 4.0) contexts: while in a domotic context IoT networks are used to provide connectivity to common and security devices (light bulbs, internal cameras, fire sensors, etc.), in an Industry 4.0 scenario, IoT is used to monitor, control, inform, and automate production processes. In order to communicate on the network, IoT devices support different communication protocols, such as Industrial Ethernet [1], Wi-Fi [2], ZigBee [3] and Z-Wave [4].

Our research, presented in this paper, investigates security aspects of IoT networks. We focus on ZigBee, a communication protocol ensuring low power consumption and characterized by low data transmission rates. During our study, we found important security issues related to a ZigBee based system and, potentially, to other IoT protocols. We identified the possibility to send Remote AT

Commands, AT meaning "attention", to a connected sensor, in order to reconfigure the device, for instance, by making it join a different malicious network, hence forward captured data to the enemy. We evaluate the possibility to perpetrate a successful attack by setting up a network laboratory composed of XBee devices (XBee is one of the most adopted ZigBee radio modules in the Do-It-Yourself (DIY) scenario [5]). We describe the exposure to Remote AT Commands threats by focusing on evaluating efficiency and performance characteristics of this innovative attack.

The focus of our work is on the proposal of an innovative cyber-attack. This may result an unconventional and not needed activity. Nevertheless, especially in the research field, it is well known that offence research is as needed ad defence one, in order to properly master a field and better prepare to counter cyber-criminal activities [6, 7].

The remaining of the paper focuses on the presentation of the innovative discovered threat and it is structured as follows: Section 2 reports the structure of the ZigBee protocol. Section 3 reports related work on the topic, while Section 4 reports our contribution on Remote AT Command exploitation. Then, Section 5 exposes the adopted testbed and obtained results by executing the attack on a controlled environment. Finally, Section 7 concludes the paper and reports possible extensions of the work.

2 The ZigBee Protocol

ZigBee is a wireless standard introduced by the ZigBee Alliance in 2004. It is based on the IEEE 802.15.4 standard, used in the Wireless Personal Area Networks (WPAN) context [8]. ZigBee is designed for embedded systems, often characterized by extremely low power consumption and low-rate transfers requirements [9]. The protocol is indeed able to minimize battery replacement frequency (up to 2 years) and to provide a communication rate up to 250 kbps, for a coverage radius up to 1000 meters. Figure 1 depicts the ZigBee stack protocol.

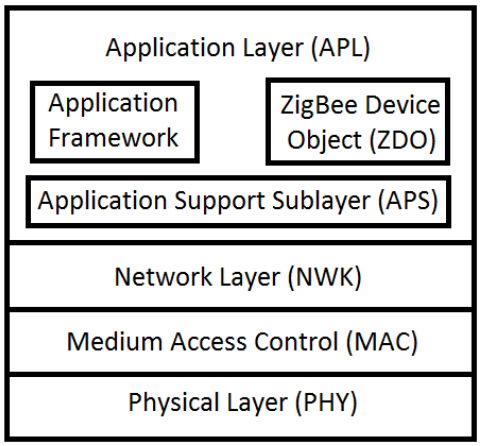


Figure 1: The ZigBee stack protocol

The physical layer of the IEEE 802.15.4 standard manages modulation and demodulation operations. Particularly, ZigBee supports three different frequencies:

- 2.4 GHz with support to 16 different channels and providing a maximum communication rate of 250 kbps (used worldwide);
- 868 MHz with support to 1 channel and a maximum data rate of 20 kbps (used in Europe);
- 915 MHz with support to 10 channels and 40 kbps of communication rate (used in US).

Since they work on the same frequency, in case of 2.4 GHz adoption, there may be interferences with existent Wi-Fi networks [10].

The MAC layer, also implemented in IEEE 802.15.4, takes care of ensuring a reliable and secure communication, by implementing a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to control access to the physical level [11].

The network layer of the ZigBee protocol implements instead network topologies, new devices management and security handling. Particulary, ZigBee supports three different network topologies:

- a star topology, where each node communicate with a central node;
- a tree topology, where central nodes of different networks are connected with a bus network;
- a mesh topology, where all the nodes are connected to each others.

Mesh networks are the most interesting ones: in this case, ZigBee implements ad-hoc routing algorithms to automatically rearrange communications if a node of the network is disconnected [12].

The application framework layer represents the user interface and it is composed by three main components:

- Application Support SubLayer (APS), providing an interface between network and application layers; moreover, it controls and manages data sent and received by other protocol layers to ensure proper packet transmission and encryption;
- ZigBee Device Objects (ZDO), an application object responsible of the initialization procedures of the APS and the ZigBee network layer to perform discovery of services and new nodes in the network;
- Application Framework (AF), an execution environment for "application objects", each ones identified by an endpoint address from 1 to 254 (0 is reserved for ZigBee Device Object (ZDO), 255 for broadcast messages). Application objects are usually implemented by different manufacturers. In order to enhance products interoperability, the ZigBee Alliance has published different Application Profiles. The most common ones are home automation, smart energy, light link and green power [8].

2.1 ZigBee Node Types

ZigBee supports different kind of devices with different functionalities:

- ZigBee end-device (ZED): it represents the sensor, usually in sleep mode most of the time and periodically waking up in order to communicate with the other nodes of the network;
- ZigBee Router (ZR): an optional node used to route packets on the network;
- ZigBee Coordinator (ZC): a ZigBee Router with gateway functions used to manage the network.

While on the same network it is common to have several different ZED nodes and different routers, a single coordinator is found.

2.2 ZigBee Security

As many other wireless networks, like Wi-Fi [13] or ad-hoc wireless sensor network [14, 15], security assumes a crucial role in the ZigBee protocol. The encryption algorithm used in ZigBee is Advanced Encryption Standard (AES) with a 128 bit key. Such algorithm, considered extremely secure and reialble, guarantees confidentiality and authenticity on a wireless communications [16].

ZigBee provides two different security profiles [17]: Standard Security, the basic security profile, rarely adopted because of its exposure to attacks, and High Security, mostly used since it guarantees greater security during communications. Particularly, while considering the Standard Security profile, the network key is shared in clear text (unencrypted), it is encrypted with the Link Key in case of High Security profile adoption. The Link Key is one of the security keys adopted by ZigBee:

- Master Key, usually hardcoded on the device or shared out-of-band, it is needed in order to retrieve the other keys but it is never directly sent on the network;
- Network Key, a key shared by all the devices connected to the same network. It is generated by the Trust Center and it can be sent on the network as plain text or in encrypted form, depending on the adopted security profile;
- Link Key, a key generated using the Master Key and adopted for communications between two different devices on the same network.

In a ZigBee network, if communication is unencrypted, an attacker may access all information of the network and may even sniff/capture exchanged packets. Otherwise, if communication is encrypted, a malicious user may only perform attacks that don't require access to the network, such as denial of service or jamming, since it is very difficult to retrieve the ZigBee adopted network key, hence decrypt exchanged packets.

3 Related Work

Because of the wide adoption of the ZigBee protocol, one of the most important concerns is related to ZigBee based networks protection. Many security experts have studied the protocol and identified several threats able to target such systems. In this context, an important contribution is provided by Joshua Wright, the creator of Killerbee, a framework including a set of tools able to exploit the ZigBee protocol analyzing network traffic and processing the recovered packets [18]. Although such software is extremely dangerous, its specific hardware requirements (such as Atmel AVR USB Stick or TelosB mote models) limit the execution to properly equipped attackers. Thanks to Killerbee, it's possible to execute several attacks against a ZigBee network: for instance, it is possible to retrieve the network key when sent as clear text. Such retrieval requires the attacker to be located in proximity of the network nodes, in order to sniff the key exchange. Killerbee also includes other threats such as replay [19] or manipulation/injection [20].

Other attacks focus on Denial of Service (DoS) activities, executed in order to disconnect a node from the network. DoS attacks are popular on the Internet [7] and they are extending to last generation fields such as mobile [21], SDN [22] and IoT [23]. Considering such kind of threats perpetrated against a ZigBee system, several attacks target battery powered sensors in order to reduce the lifetime of the device. In this context, the ZigBee end-device Sabotage attack [24] is executed by keeping sensors active through the send of a broadcast message every time the device wakes up from the sleep status. In this way, a sensor under attack is forced to reply the malicious user, hence delaying the next sleep and discharging the batteries quickly. A similar threat, the ghost attack proposed by Devu Manikantan Shila et al., reduces the lifetime of the targeted device by sending several crafted bogus messages to the victim [25]. Niko Vidgren et al. demonstrate instead how it is possible to discharge the batteries of a sensor if the attacker knows the adopted sensor polling rate [26]. Pacheco et al. investigate instead DDoS attacks feasibility against IoT environments [27]. Another DoS attack proposed by Niko Vidgren et al. exploits the ZigBee frame counter. Such counter is commonly used by different network protocols to prevent threats such as replay attacks. Concerning ZigBee frame counter exploitation, a malicious user could send a parameter containing the maximum allowed frame counter value (sized 4 bytes), hence forcing the victim to set the counter to the received value. If Message Integrity Check [28] is

not implemented by the victim, each packet received after the malicious one will be discarded by the victim since it will present a lower frame counter [26].

Another attack, known as Same-Nonce attack [29], can be carried out only if the Trust Center, a device providing reliability during the key exchange stage, provides the same nonce encrypt with the same network key for two consecutive times. In a ZigBee network, coordinators has role of Trust Center. In this scenario, an attacker may retrieve part of the plain text simply calculating the XOR between the two sniffed packets. Although this situation rarely happens, it is possible to force this behavior by causing a power failure, e.g. by discharging batteries of a Trust Center. In this case, Trust Center resets the nonce to its default value and it is possible to send a packet with the same nonce [26].

Considering other threats, ZigBee networks are also vulnerable to attacks known as Sinkhole and Wormhole, proposed by Azmi Bin Karnain et al. [30]. During a Sinkhole attack, a malicious node attracts the network packets with the aim of creating confusion in the routing phase. Instead, during a Wormhole attack, the malicious user receives packets at one point in the network and then replays this packets in other areas to interfere all network functionality. Also, while Irina Krivtsova et al. proposes the Broadcast Storm attack clogging the network by sending numerous broadcast packets [31], Wei Yang et al. introduces two attacks against ZigBee, known as Absolute Slot Number (ASN) and time synchronization tree attack. Considering that the time is splitted in different slots/ASNs of fixed length, during an ASN attack, since current ASN value is sent during communication, the legitimate nodes may get an incorrect ASN value from the attacker that sends on the network a broadcast message with a wrong ASN value. In this way, a node wouldn't be able to communicate on the network since the wrong ASN packet would lead to a communication interruption. In time synchronization tree attack, the malicious user may send bogus DAG Information Objects (DIO) packets [32] to the neighbors with the aim to de-synchronize their connections with the network [32]. A sybil attack, proposed by Gunhee Lee et al., is launched by an attacker that acquires multiple identities on the network. The aim of this attack is to convince the other devices that the malicious node is a legitimate node. In this way, a malicious node may, e.g., access all services of the network or identify itself as a ZigBee router [33]. Another type of attack is performed if the enemy can physically access to a ZigBee device. Indeed, the malicious user may perform a firmware dump in order to extrapolate the network key stored/hard-coded in the device [34].

Other attacks focus instead on specific version of ZigBee. In this context, a particular version of the ZigBee protocol, called ZigBee Light Link, used for instance by Philips Hue bulbs [35], has been exploited different times. Indeed, Colin O'Flynn found that the adopted ZigBee network key can be retrieved if an attacker can sniff the re-initialization process accomplished by the bulbs after a reset and if he knows the ZigBee Light Link master key [36]. Another attack on ZigBee Light link is proposed by Eyal Ronen and Colin O'Flynn, creating a worm that automatically infects adjacent bulbs, building a custom infected firmware and to be deplyed as a fake OTA update [37].

Many works focus instead on ZigBee protection. For instance, a solution to detect the sybil attack is proposed by Salavat Marian et al., presenting an interesting protection system using RSSI derived metris to detect a sybil attack by computing the location of a node and then classifying it as malicious or not [38]. Bilal AI Baalbak et al. introduces instead an Anomaly Behavior Analysis System (ABAS) for the ZigBee protocol based on network traffic analysis. After a detection is triggered, ABAS can classify the attack as known or unknown using information like packets origin or destination [39]. Another protection algorithm proposed by Paria Jokar et al. and known as HANIDPS, implements a machine learning based intrusion detection and prevention system. HANIDPS analyzes the network traffic an compares it with a normal in order to detect a running threat [40]. A similar approach may analyze energy consumption [41] to identify running attacks. Baojiang Cui et al. proposes instead a fuzzing method based on finite state machines. A fuzzy test is implemented by injecting different testing cases into the system in order to detect vulnerabilities [42]. A defense against impulsive noise

is proposed by Jia Jia et al., implementing a system using a noise filtering processing activity in two steps: while during the first step an estimate of the noise is computed, in the second one is a noise cancellation is accomplished, in order to state if the estimate is suspect or not [43].

During our research work, we have studied security aspects of ZigBee based IoT networks by initially studying the protocol, thus analyzing the major threats affecting it, hence studying possible protection systems and approaches. During our study, we have discovered the proposed threat and, to the best of our knowledge, we noticed that a vulnerability analysis focused on AT Command exploitation is still missing. Nevertheless, this vulnerability should be considered extremely innovative and particularly dangerous, since it allows malicious users to retrieve/forward sensitive information or manipulate nodes functionality. Our work focuses on the proposal of the innovative Remote AT Command attack, explained in the next section, by illustrating the proposed threat and evaluating its efficiency.

4 Remote Control Exploitability

In order to properly investigate ZigBee security, we have studied the protocol and analyzed communication flows, considering the different types of packets supported by ZigBee. While, at first, we focused on packets containing data sent from the coordinator to the end-device, later, we have also analyzed other packets exchanged in the network. In this context, we found that, at the MAC layer, it is possible to send Remote AT Commands. By working at such lower layer, received packets are not processed at the application layer, hence it may not be not possible to access to the packet content to avoid interpretation, except from the device manufacturer.

Our work focuses on AT Commands exploitation, a particular vulnerability we have identified and affecting IoT sensor networks. AT Commands are specific packets, historically adopted by old generation modems to interface with the device, today used by radio modules such as XBee [44], ESP8266 [1] or ETRX3 [45] to configure parameters like connection type, network identifier, device name on the network or destination address for a communication. AT Commands are today supported by many devices of different nature and providing different functionalities, hence commands. For instance, modules that provide connectivity support AT Command packets for network parameters configuration while other modules may use these packets to alter light intensity of light bulbs.

For our research, as previously mentioned, we focused on XBee modules. Such modules, widely adopted around the world, especially in DIY contexts, implement two different AT Command packets, related to request and response operations, respectively. Concerning XBee modules, these packets can be sent remotely: we talk in this case of Remote AT Commands. Such packets belong to the (IEEE 802.15.4) MAC layer and they are interpreted by the (XBee) module automatically. Therefore, by being such interpretation demanded to the device firmware, and being such firmware provided by the manifacturer, Digi International, it is not possible to avoid implict Remote AT Commands interpretation. In order to execute the proposed attack, the AT command functionality of XBee has to be exploited. XBee supports several AT Command packets [2]. Particularly, for our aim, we have used ATID commands to target sensors (in general, other commands/approaches may be used for different purposes: e.g. to make the sensor join a different network, to forward (sensitive) data to a different malicious receiver, to disable data encryption, etc.). ATID is used by XBee modules to set the network identifier. During the proposed Remote AT Command attack, the malicious user sends an ATID packet with a bogus identifier in order to make it join a different (inexistent, in our case) network.

In order to maliciously exploit Remote AT Command, it is assumed that the attacker is connected to the network of the target. In this case, the enemy may, for instance, disconnect an end-device

[1] More information are available at: `http://esatjournals.net/ijret/2017v06/i01/IJRET20170601027.pdf`

[2] More information are available at `https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf`

from the ZigBee network and make it join a different (malicious) network, hence forward potentially sensitive data to third malicious parties. Given the nature of IoT end-devices, often associated to a critical data and operations, it may be obvious how to a Remote AT Command attack represents a serious threat for the entire infrastructure.

5 Testbed

In this section, we report information about the tests we have conducted in order to validate the success and the efficiency of a Remote AT Commands attack. In order to accomplish the tests, we have built a ZigBee test network, depicted in Figure 2.

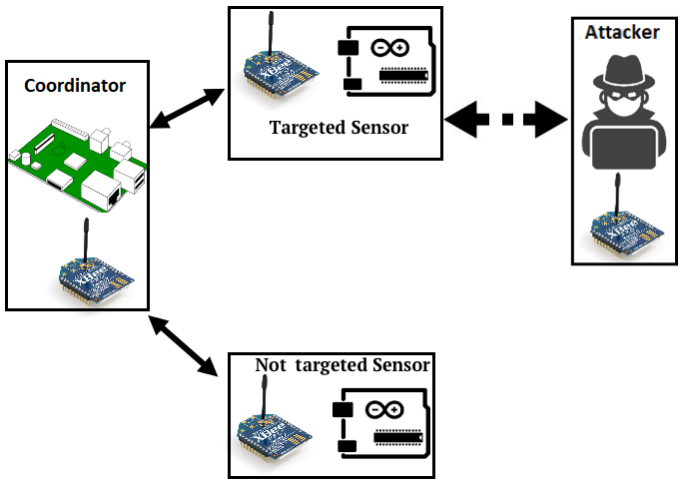


Figure 2: Test network

The network is composed of a single ZigBee coordinator, two end-devices representing common sensors on the network, and a malicious user/node connected to the ZigBee network. As can be deduced from the figure, the attacker sends Remote AT Command packets only to one sensor and not to each device on the network. This implementation allows us to monitor the effects of the attack on the two sensors, hence evaluating the possibility to carry out a successful attack without affecting not targeted nodes. Indeed, we expect that during a Remote AT Command attack, only the targeted sensor in affected by the threat, while other nodes keep working correctly (unless their behaviour depends on the targeted sensor).

Considering the described scenario, we will now detail at first adopted hardware, hence reporting information about testbed configuration, finally exposing the obtained results.

5.1 Testbed configuration

Different devices have been used to create ZigBee network to implement AT Command attack. For our aim, network components are composed as reported in the following:

- *Coordinator*, composed of a Raspberry Pi 3 equipped with an XBee USB Board and an XBee Series 2;
- *Targeted Sensor*, composed of an Arduino UNO R3, equipped with an XBee Shield and an XBee Series 2;
- *Not targeted Sensor*, composed of an Arduino UNO R3, equipped with an XBee Shield and an XBee Series 2;

- *Attacker*, composed of a Raspberry Pi 3 equipped with an XBee USB Board and an XBee Series 2.

As the reader may notice, end-devices/sensors share the same hardware. Hence, our evaluation allows us to identify the efficiency of the attack on the targeted node, and simultaneously the possibility to avoid side effects on other nodes (this is not possible, e.g., for jamming attacks).

Moreover, since XBee series 2 modules have low computational capacity, we adopted Arduino microcontrollers to generate and elaborate information, hence using XBee modules only for network communications. In order to guarantee the sleep status of end-devices, a PIN Hibernation has been implemented [46] by connecting the 7^{th} PIN of the Xbee Shield to an Arduino digital PIN. In order to implement PIN Hibernation, $D7$ value, a PIN used to send and receive serial data, has been disabled (through XCTU XBee programming software). In order to test this vulnerability, the innovative attack is implemented and tested with this configuration: the attack was performed on only one sensor because by monitoring the network traffic is possible to verify the efficiency of this threat.

5.2 Network Nodes Implementation

Every 35 seconds, sensors are programmed to send a packet to the coordinator. Each packet contains a random generated number. After the message has been sent, the sensor device enters in sleep mode in order to reduce power consumption. Since the content of the message is not meaningful to us, the "random number" solution allows us to generate data to be transmitted on the network without requiring environmental sensors.

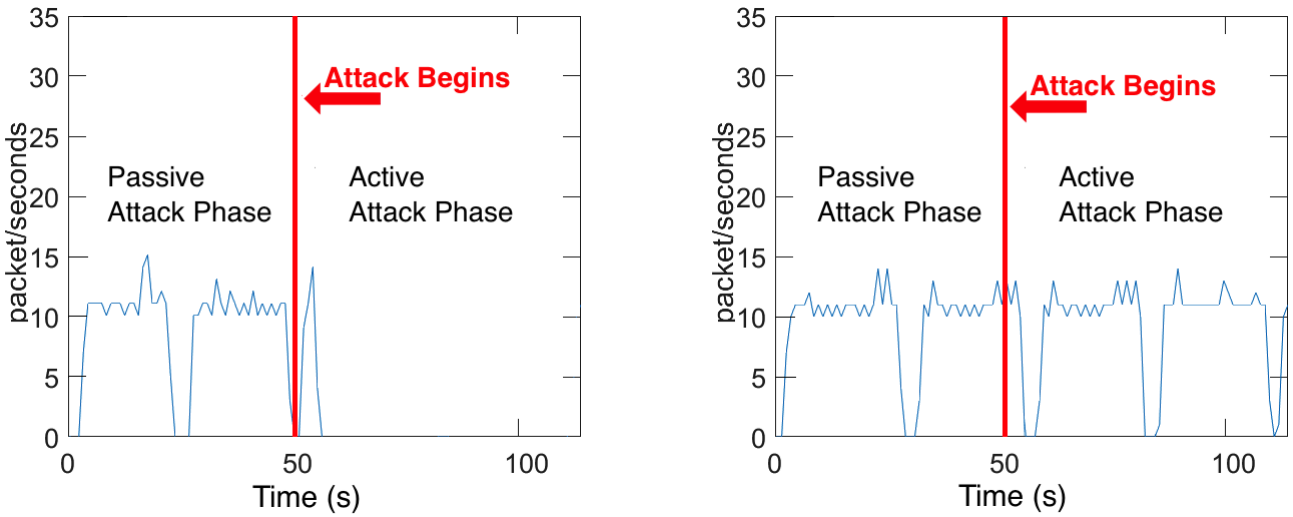
Figures 3 and 4 monitor the network traffic of the various XBee modules. Although we stated that a single packet is sent every 35 seconds, such send is relative to application layer packets, while the capture is relative to the entire ZigBee network stack. Although such capture includes additional (lower layers) packets (including, for instance, wake-up commands containing network node information, synchronization packets, etc.), it is representative of the network behaviour of the sensor (e.g. we can see that after the attack, no packets are sent by the victim node), while a capture focused on application layer packets/messages would produce single peaks missing useful information.

Data is received by the coordinator and shown to the user trough an HTML based graphical interface, also reporting if sensors are correctly communicating with the coordinator. This environment is representative of a wide range of network types. For instance, sensors installed on a specific area/farm/company could be monitored through a similar approach, or intrustry machines and fire prevention systems may be part of a network system similar to the proposed one.

6 Results

Network traffic was analyzed from an external ZigBee device capturing data on the same channel used by the targeted network. From sniffed traffic, we are able to extrapolate communication flows of single hosts of the network.

Figure 3a and Figure 3b report the network traffic flow of both targeted and not targeted sensors during a running attack. Traffic was monitored for 120 seconds and it is splitted in two phases: during the first 50 seconds, the attacker acts in a "passive" way, by scanning the ZigBee spectrum in order to identify the devices connected to the network and define the target. Instead, on the second "active" phase, the attacker sends Remote AT Command packets to the targeted sensor in order to perform the attack. Particularly, for our aim, the passive behavior is not intended as a "listen only" behavior. Instead, during this phase, the attacker does not send any malicious packet on the network. Therefore, we expect that detecting a malicious behavior during the passive phase is particularly difficult.



(a) Traffic generated by targeted sensor

(b) Traffic generated by not targeted sensor

Figure 3: Network Traffic Captured during attack execution

Figure 3a reports the traffic flow of the targeted sensor. By analyzing the graph, it is possible to notice that while the sensor is correctly working during the passive phase, a few seconds after the attack is (actively) performed, the device is disconnected from the network and its communication with the coordinator is interrupted. Therefore, the attack results successful on the targeted sensor.

Instead, Figure 3b reports the status of the non targeted sensor during the attack. Particularly, it is possible to notice that the connection is maintained alive for the entire considered period. Indeed, since this sensor is not directly targeted by the attacker, Remote AT Commands are not received/interpreted, hence, the network parameters of the sensor are not altered by the attacker and communication capability of the sensor is maintained and not even disturbed. This represents an important characteristic of the proposed threat, since it is able to only affect the targeted device, by making the attack not directly visible to the other sensors/devices. Such stealth behavior makes the attack more difficult to detect. Moreover, considering that device communication interruption may be related to external factors (e.g., battery drain, wireless noise, malfunctioning device, etc.), the proposed attack should be considered a serious threat.

Figure 4 shows instead the captured attack traffic during the considered period.

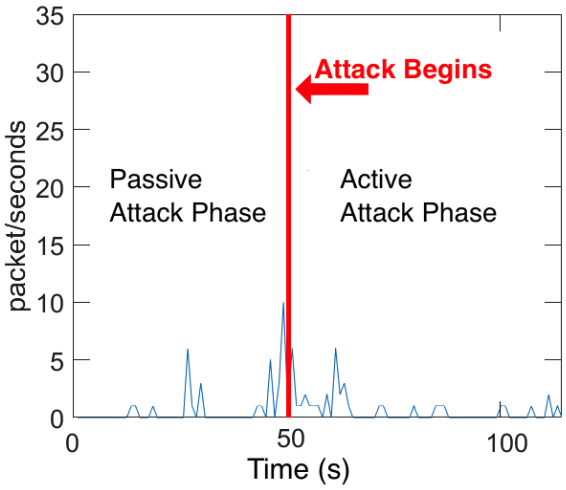


Figure 4: Traffic generated by attacker sensor

By analyzing the passive phase of the attack, as previously mentioned, the enemy performs a scan of the the network in order to identify each device connected to the network and choose the targeted device. Instead in the active attack phase, Remote AT Command packets are sent to the targeted sensor with the aim of disconnecting it from the network (by reconfiguring it). If we analyze the attack traffic flow, it is very difficult to distinguish a passive (hence, potentially legitimate) behavior from an active (malicious) one. Hence, detection of a running threat may require packet inspection or data flows interpretation (not easy to accomplish in case of encrypted traffic).

Although our testbed focus on two network sensors/devices, the proposed Remote AT Commands attack results particularly scalable, due to the (minimum) requirements for the attacker (a single packet is sent to the victim to reconfigure it). Particularly, the time required to send such packet is minimal, so in case of multiple targeted sensors, the attack success is guaranteed. Of course, in case of extremely large amounts of targeted sensors, the effectiveness of the attack depends on the scan time: the attack is successful if the minimum "sleep time" of each sensor is larger than the average time required to target all the sensors.

7 Conclusion and Future Work

The proposed paper is focused on Internet of Things (IoT) environments security, by analyzing the possibility to carry out a successful attack against a targeted node/sensor/device. During our work, we found a novel vulnerability affecting IoT devices: by exploiting a particular type of packet, Remote AT Command, it is possible to remotely reconfigure/program network nodes as the attacker wishes, hence compromising data communication security of the network.

By focusing on the ZigBee (wireless) protocol, we have described and implemented the proposed attack with the aim to interrupt the communication capabilities of a targeted device of the network. For our tests, we targeted XBee modules [44], able to communicate through the ZigBee protocol. Results show that the attack is successful and it is able to target a single node without affecting the other nodes of the network. Moreover, since the number of packets sent by the attacker is miminum, it is not easy to detect a running attack, without doing deep packet inspection. The attack results therefore particularly dangerous, since it may compromise the security of an IoT network with minimum effort for the attacker. By comparing the effects of the proposed attack to other network based threats, they can be assimilated to denial of service, man-in-the-middle (traffic sniffing), or traffic redirection activities, in function of the strategy adopted by the attacker.

Future work on the topic may concern additional tests of the attack in large scale networks composed by different nodes, in order to identify the limits of the threat, in function of the sleeping/polling times adopted by the nodes. Considering instead the design of defense systems, additional extensions of the work may be directed to the implementation of efficient protection techniques able to defend an IoT system from a Remote AT Command attack. Since detection of a running threat may not be immediate, in order to protect a remote device from a Remote AT Command attack, it may be preferred to directly work on the (potentially vulnerable) nodes. In this context, three different approaches can be adopted, working at different levels:

- Firmware level: creation of a modified version of the firmware, implementing Remote AT Commands filtering or allowing AT Commands elaboration at the application layer;
- Device configuration level: providing to the user the ability to configure a device with disabled support to Remote AT Commands;
- External level: demanding protection capabilities to an external application program.

Each approach provides an efficient solution to protect the device. Nevertheless, some approaches may not be adopted (e.g. device configuration, if not available). Suggested implementations provide a possible protection for this innovative threat.

The first proposed solution (firmware level protection) requires a device firmware upgrade to allow total AT Command packet management, such as the ability to process only packets received by the coordinator or secure devices. Such solution would provide the user the possibility to configure the device in order to avoid implicit Remote AT Commands interpretation.

Since modifying a firmware may not be easy, and the source code must be open source, it is suggested to have simpler but equally effective solutions. The second solution (device configuration level protection) implements the ability to disable Remote AT Command support of the module, by implementing a specific settings able to disable automatic Remote AT Command interpretation (e.g. packets discard). In this way, the proposed threat would be ineffective.

The last proposed solution (external level protection) is the most interesting, the main purpose is to implement protection logics on the Arduino device by implementing a function at Application Layer. The aim of the function is to verify if the XBee module may be communicate on the network. In this case, just before the sensor is ready to communicate on the network, an internal check is accomplished.

Although the mentioned approaches may protect IoT modules and network sensors from this innovative attack, by ensuring data transmission security, their design implementation and evaluation is on the scope of further work on the topic.

Acknowledgment

This work has been supported by the following research projects:

- My Healt-My Data (MHMD) project has received funding from the European Union's Horizion 2020 research and innovation programme under grant agreement No 732907
- Advanced Networked Agents for Security and Trust Assessment in CPS/IoT Architectures (ANASTACIA) project has received funding from the European Union's Horizion 2020 research and innovation programme under grant agreement No 731558

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] J. S. Rinaldi and P. S. Marshall, "Industrial ethernet," *ISA Press Release*, vol. ISSN: 978-1-945541-04-9, 2004.

[2] e. a. L. Li, "The applications of wifi-based wireless sensor network in internet of things and smart grid," pp. 789–793, 2011.

[3] e. a. C. Fan, "A middleware of internet of things (iot) based on zigbee and rfid," pp. 732–736, 2011.

[4] Z. W. Alliance, "The internet of things is powered by z-wave.(2016)," *Z-Wave Alliance*, vol. 28, p. 2016, 2016.

[5] R. Faludi, *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing.* " O'Reilly Media, Inc.", 2010.

[6] e. a. P. Farina, "Are mobile botnets a possible threat? the case of slowbot net," *Computers & Security*, vol. 58, pp. 268–283, 2016.

[7] e. a. E. Cambiaso, "Slow dos attacks: definition and categorisation," *International Journal of Trust Management in Computing and Communications*, vol. 1, no. 3-4, pp. 300–319, 2013.

[8] e. a. C. M. Ramya, "Study on zigbee technology," vol. 6, pp. 297–301, 2011.

[9] e. a. A. Dementyev, "Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario," pp. 1–4, 2013.

[10] e. a. M. A. Sarijari, "Experimental studies of the zigbee frequency agility mechanism in home area networks," pp. 711–717, 2014.

[11] e. a. P. Baronti, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards," *Computer communications*, vol. 30, no. 7, pp. 1655–1695, 2007.

[12] e. a. J. Li, "Study on zigbee network architecture and routing algorithm," vol. 2, pp. V2–389–V2–393, 2010.

[13] S. Gold, "Cracking wireless networks," *Network Security*, vol. 2011, no. 11, pp. 14–18, 2011.

[14] E. Cayirci and C. Rong, *Security in wireless ad hoc and sensor networks.* John Wiley & Sons, 2008.

[15] L. Caviglione and F. Davoli, "Peer-to-peer middleware for bandwidth allocation in sensor networks," *IEEE communications letters*, vol. 9, no. 3, pp. 285–287, 2005.

[16] e. a. L. Caviglione, "Measuring the energy consumption of cyber security," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 58–63, 2017.

[17] G. Dini and M. Tiloca, "Considerations on security in zigbee networks," pp. 58–65, 2010.

[18] J. Wright, "Killerbee: practical zigbee exploitation framework," 2009.

[19] B. Stelte and G. D. Rodosek, "Thwarting attacks on zigbee-removal of the killerbee stinger," pp. 219–226, 2013.

[20] e. a. A. Biswas, "A lightweight defence against the packet in packet attack in zigbee networks," pp. 1–3, 2012.

[21] e. a. R. H. Jhaveri, "Dos attacks in mobile ad hoc networks: A survey," pp. 535–541, 2012.

[22] R. Kandoi and M. Antikainen, "Denial-of-service attacks in openflow sdn networks," pp. 1322–1326, 2015.

[23] e. a. H. Suo, "Security in the internet of things: a review," vol. 3, pp. 648–651, 2012.

[24] e. a. O. Olawumi, "Three practical attacks against zigbee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned," pp. 199–206, 2014.

[25] e. a. D. M. Shila, "Ghost-in-the-wireless: Energy depletion attack on zigbee," *arXiv preprint arXiv:1410.1613*, 2014.

[26] e. a. N. Vidgren, "Security threats in zigbee-enabled systems: vulnerability evaluation, practical experiments, countermeasures, and lessons learned," pp. 5132–5138, 2013.

[27] e. a. L. A. B. Pacheco, "Evaluation of distributed denial of service threat in the internet of things," pp. 89–92, 2016.

[28] e. a. H. Li, "Application and analysis of zigbee security services specification," vol. 2, pp. 494–497, 2010.

[29] N. Sastry and D. Wagner, "Security considerations for ieee 802.15. 4 networks," pp. 32–42, 2004.

[30] M. A. B. Karnain and Z. B. Zakaria, "A review on zigbee security enhancement in smart home environment," pp. 1–4, 2015.

[31] e. a. I. Krivtsova, "Implementing a broadcast storm attack on a mission-critical wireless sensor network," pp. 297–308, 2016.

[32] e. a. W. Yang, "Security vulnerabilities and countermeasures for time synchronization in ieee802. 15.4 e networks," pp. 102–107, 2016.

[33] e. a. G. Lee, "An approach to mitigating sybil attack in wireless networks using zigbee," vol. 2, pp. 1005–1009, 2008.

[34] L. Jun and Y. Qing, "Take unauthorized control over zigbee devices, https://media.defcon.org/def con 23/def con 23 presentations/defcon-23-li-jun-yang-qing-i-am-a-newbie-yet-i-can-hack-zigbee.pdf," 2015.

[35] J. Wang, "Zigbee light link and its applicationss," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 6–7, 2013.

[36] C. O'Flynn, "A lightbulb worm?, https://www.blackhat.com/docs/us-16/materials/us-16-oflynn-a-lightbulb-worm-wp.pdf," *Blackhat*, 2016.

[37] e. a. E. Ronen, "Iot goes nuclear: Creating a zigbee chain reaction," pp. 195–212, 2017.

[38] S. Marian and P. Mircea, "Sybil attack type detection in wireless sensor networks based on received signal strength indicator detection scheme," pp. 121–124, 2015.

[39] e. a. B. Al Baalbaki, "Anomaly behavior analysis system for zigbee in smart buildings," pp. 1–4, 2015.

[40] P. Jokar and V. Leung, "Intrusion detection and prevention for zigbee-based home area networks in smart grids," *IEEE Transactions on Smart Grid*, 2016.

[41] e. a. L. Caviglione, "Seeing the unseen: revealing mobile malware hidden communications via energy consumption and artificial intelligence," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 799–810, 2016.

[42] e. a. B. Cui, "A novel fuzzing method for zigbee based on finite state machine," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[43] J. Jia and J. Meng, "A novel approach for impulsive noise mitigation in zigbee communication system," pp. 1–3, 2014.

[44] R. Piyare and S. r. Lee, "Performance analysis of xbee zb module based wireless sensor networks," *International Journal of Scientific & Engineering Research*, vol. 4, no. 4, pp. 1615–1621, 2013.

[45] A. T. C. Dictionary, "Etrx2 and etrx3 series zigbee® modules at-command dictionary, https://www.silabs.com/documents/public/reference-manuals/tg-etrxn-commands.pdf," 2010.

[46] P. Manual, "Xbee/xbee-pro rf modules, http://store.express-inc.com/pdf/xa-a.pdf,"