Security issues in internet of things: Vulnerability analysis of LoRaWAN, sigfox and NB-IoT

Coman, Florian Laurentiu; Malarski, Krzysztof Mateusz; Petersen, Martin Nordal; Ruepp, Sarah Renée

Published in:
Proceedings of 3rd Global IoT Summit

Link to article, DOI:
10.1109/giots.2019.8766430

Publication date:
2019

Document Version
Peer reviewed version

Security Issues in Internet of Things: Vulnerability Analysis of LoRaWAN, Sigfox and NB-IoT

Florian Laurentiu Coman, Krzysztof Mateusz Malarski, Martin Nordal Petersen and Sarah Ruepp
Technical University of Denmark, 2800 Kongens Lyngby, Denmark
lauroflorin@gmail.com, {krmal, mnpe, srru}@fotonik.dtu.dk

***Abstract*—One of the key enablers of an increased Internet of Things (IoT) roll-out is Low-Power Wide Area Network (LP-WAN) - a family of technologies tailored for resilient and energy-efficient communication of thousands of devices over large distances (even up to 100km). Under the pressure from both the business and the society to provide ubiquitous connectivity as soon as possible, new IoT deployments are conducted with haste and often by inexperienced people. Consequently, the aspect of communication security traditionally remains a secondary matter, even though the potential harm of a successful hacker attack can be enormous. Therefore, this paper presents an analysis of LP-WAN vulnerabilities, as well as several Proof-of-Concept (PoC) attacks toward LoRaWAN (packet forging), Sigfox (replay with DoS) and NB-IoT (attack using malicious UE), that confirm the existence of the vulnerabilities in both the standards and off-the-shelf hardware and services.**

***Index Terms*—Security, NB-IoT, LoRaWAN, Sigfox, vulnerability, LP-WAN.**

I. Introduction

Although Internet-connected everyday devices have already been present for a long time, recently, a new class of IoT technologies have emerged: LP-WAN technologies. As the name suggests, LP-WAN focuses on covering a large physical area while using as little battery power as possible. Achieving both of these may seem counter-intuitive, but LP-WANs are made possible by using lower frequencies (attenuation rate increases with frequency), lower bit rates and methods that improve robustness such as repetition schemes or spread spectrum techniques (such as Chirp Spread Spectrum (CSS) or Frequency Hopping Spread Spectrum (FHSS)).

Some of the most well-known members of the LP-WAN family are Sigfox[1], Long Range Wide Area Network (LoRaWAN)[2] and Narrowband-IoT (NB-IoT)[3]. While NB-IoT is a Cellular-IoT standard, meaning it uses licensed spectrum as other mobile technologies (2G,3G,4G), LoRaWAN and Sigfox both operate in unlicensed spectrum - the Industrial, Scientific and Medical (ISM) band. We consider the aforementioned connectivity options, as they are already deployed or being deployed in many countries, and the compliant hardware is commonly available.

LP-WAN technologies can be employed in some critical applications, such as asset tracking, or remote monitoring of e.g. rail temperature. In such scenarios, both reliability and security guarantees of the data and communication are essential. A successful attack on data transmission may lead to financial losses (corrupted or modified meter readings) and a danger to the people (malicious functioning of city facilities or even a train accident). Moreover, if a hacker compromises IoT devices and uses them as computational resources for another malicious action (i.e. if he/she creates a botnet), then the capabilities of a Distributed Denial-of-Service (DDoS) attack will skyrocket; e.g. Mirai botnet, consisting of tens of millions of devices, reached 1.2 Tbits/s[4]. However, the vulnerability of LP-WAN may also reside in the core network, for example, erroneously implemented charging function in the cellular network can lead to overcharging attacks toward NB-IoT subscribers [5]. Therefore, in this paper we present our findings on security issues in three most common LP-WAN technologies: LoRaWAN, Sigfox and NB-IoT.

978-1-7281-2171-0/19/$31.00 © 2019 European Union

Related work and contributions

The research on LP-WAN security vulnerabilities done so far has been primarily focused on LoRaWAN. In [6], 5 attacks on LoRaWAN 1.0.x are analysed: replay, eavesdropping, packet modification, ACK spoofing and battery exhaustion. The authors implemented PoC experiments of the attacks, and suggested improvement steps. A detailed and extensive study on security aspects of LoRaWAN 1.1 is presented in [7], however the authors did not employ a test-bed with real hardware. As far as Sigfox or NB-IoT is concerned, their security issues are only briefly mentioned in [8] and [9]. The authors of [9] attempt to provide an overview of security issues in LPWAN networks (LoRaWAN, NB-IoT, Sigfox, DASH7), however the analysis provided is very superficial and no results from hands-on experiments are presented. Thus, security issues of LPWAN need to be further studied.

The main contributions of this paper are the following:

- We provide a thorough analysis of general security issues relevant for all LP-WAN standards.
- We introduce a selection of technology-specific vulnerabilities in LoRaWAN, Sigfox and NB-IoT, and describe the Proof-of-Concept (PoC) attacks addressing them.

This paper is organised as follows. Section II introduces generic security threats in LP-WAN. Section III presents technology-specific security weak-points and the corresponding attacks implemented. Section IV concludes the work.

II. LP-WAN security threats

In this section we present technology-agnostic threats that correspond to different aspects of IoT communication: hard-

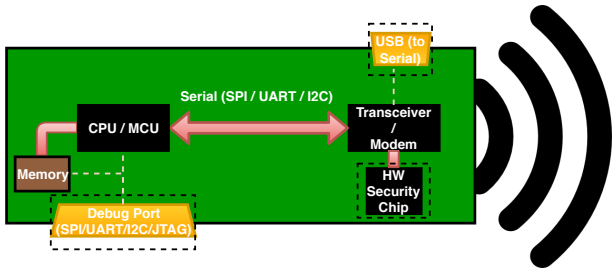

Fig. 1. Node/gateway architecture

ware, signal intelligence/traffic analysis, jamming and security keys.

A. Threats on physical security

If an attacker gets access to a node, a gateway, or a server, and if strong hardware security policies are not used, the whole device or even the network must be assumed as compromised. In the case of a node or gateway takeover, security keys could be extracted, forged messages could be sent as though originating from the node, every message passing through it can be intercepted, or the device could be destroyed. In the case of a server takeover, this could potentially lead to a whole system compromise. If security keys are stolen, this breaks both confidentiality and integrity of the device on the long term, as the attacker can intercept, decrypt or forge any messages sent within the LP-WAN system.

Fig. 1 shows a possible hardware architecture of a LP-WAN device or gateway. Hardware Security Module (HSM) contains security keys and cryptographic functions (e.g. pseudorandom number generator and encryption algorithms) and should be tamper-proof to ensure the keys are deleted when an attacker tries to extract them. If no HSM is used, it can be observed from Fig. 1 that the keys have to be kept in an unsafe storage (e.g. simple non-volatile memory) and risk being extracted by dedicated individuals. However, even the presence of HSM does not prevent the attacker from side-channel analysis that can be used to recover the secret keys, as shown in [10] which recovered the keys found in 3G/4G USIM Cards, or [11] which extracted the global AES-CCM key Philips used to encrypt and authenticate firmware on its ZigBee-connected lightbulbs.

B. Signal intelligence and traffic analysis

IoT protocols that do not offer encryption can be easily intercepted, allowing an attacker to read the application payload. Analysing how the payload is structured is part of signal intelligence. An attacker can analyse how the bytes are changing with each transmission, and try to understand how the payload looks like. Since there is a finite number of possible data structures, and given the small size of the payload in LP-WAN, this task can be completed easily. Also, by following the guidelines provided by each LP-WAN service provider, one can expect the default data structures present in the guidelines to be used by the majority of their users.

Even if a protocol offers encryption, much can be learned by analysing communication patterns (Traffic Analysis). As an example, consider the use case of sending a message whenever a door is locked/unlocked. The node would transmit a message only when a trigger is detected, e.g. when a door is locked/unlocked, with a payload of ”door locked”/”door unlocked”. Even if this payload is encrypted, resulting in an encrypted payload of e.g. ”ucerlzxc34g” and ”kcor8309gkzvv”, (a) the encrypted message is sent only when the door is locked/unlocked, allowing an attacker to know when these triggers were detected (e.g. when a user comes/leaves home) and (b) the payloads have different sizes, so an attacker can possibly differentiate between the 2 commands (e.g. if a larger payload is intercepted, this means the door was unlocked).

C. Jamming

Since LP-WAN technologies are characterised by large coverage, good link budget and multiple gateways receiving their messages, jamming might prove difficult. However, their communication bandwidth is small (100Hz for Sigfox, 125/250/500kHz for LoRaWAN, 180kHz for NB-IoT) and they use low power for transmitting, so the jammer hardware need not be complicated as long as it sends the jamming signal at a high enough power. Since a downlink message (from gateway to end node) does not have spatial diversity, it can most likely be jammed if the attacker is close to the end node. Jamming attacks may target different layers of the OSI model: 1) Physical layer jamming, where the attacker sends any wideband signal with a higher Signal-to-Noise Ratio (SNR) than the victim; 2) MAC layer jamming, where the attacker only jams specific parts of the message (e.g. the CRC32, or message signatures), ensuring the packet is discarded by the recipient.

Defending against jamming is difficult, since these attacks are always possible, but increasing the number of gateways will reduce the chance of an attacker. Jamming detection mechanisms can also be used, to e.g. change the used frequency channels and notice the local authorities when this happens.

D. Security keys

Another aspect to take into account is the size of the security keys and their cryptoperiod (length of time using the same security keys). At the moment, 128 bits for an AES key might be enough to prevent bruteforcing attempts at guessing the key, but this might not be the case in 10 years from now (which is the presumed lifetime of an LP-WAN device), due to increased computing power available in the future. National Institute of Standards and Technology (NIST) recommends a cryptoperiod of less than 5 years for every type of key (e.g. symmetric, private/public etc.) [12], which will be hard to achieve for technologies which do not allow the master key to be changed remotely (e.g LoRaWAN, Sigfox and NB-IoT).

III. LP-WAN VULNERABILITIES

In this section we motivate and present the attacks on LP-WAN systems. For each selected vulnerability we describe the

security mechanism and we identify the weak-points, which are then exploited by PoC attacks. It has to be mentioned that all the PoCs were conducted in a closed environment (e.g. Faraday Cage), and the operators and manufacturers were informed about the discovered vulnerabilities prior to the publication of this article.

A. LoRaWAN packet forging (MIC bruteforcing)

LoRaWAN uses a 4 byte Message Integrity Code (MIC) to protect against malicious actors forging packets. The MIC is calculated over the whole LoRaWAN packet and uses a 128 AES key (NwkSKey) in CMAC mode to do that. The MIC ensures packets cannot be forged by an attacker, since calculating the MIC requires knowledge of the NwkSKey. When a packet arrives to the Network Server, it discards packets that have an invalid MIC.

Theoretically, without knowing the NwkSKey, an attacker could forge any packet by bruteforcing the MIC until the correct MIC has been found. However, with a 4 byte MIC the attacker would need to send well over 4 billion LoRaWAN packets in order be sure about the correct MIC.

This, at a rate of 10 LoRaWAN packets per second, would take 13.6 years. However, taking advantage of a gateway's ability to simultaneously demodulate data received on multiple channels at once, the approach can be parallelized by sending on multiple channels (e.g. 868.1, 868.3 and 868.5MHz) and with different spreading factors (e.g. SF 7, 8, 9, 10, 11, 12 and FSK) simultaneously. An attacker could even send to multiple gateways (if they are configured to forward their messages to the same Network Server (NS)). If we assume the same sending rate of 10 packet per second (although different SFs will use different transmission speeds), 15 channels, 7 orthogonal modulation schemes (6 LoRa SFs and 1 FSK), and 5 gateways, the bruteforcing will take less than 10 days. With a device lifetime of more than 10 years, this approach might prove useful for some cases. Yet, what really enables such an attack is switching to IP domain: sending an IP packet is orders of magnitude faster than sending a LoRaWAN packet. Hence, if the attacker can send 4 billion packets directly to the NS, this attack becomes possible. Tools such as masscan or Zmap can scan the whole IPv4 internet address space (also 4 bytes as the LoRaWAN MIC) on one port in less than 6 minutes, which indicates a sending rate of around 12 million packets/second, so this attack is plausible. This approach can also be parallelized by sending from multiple sources, the only bottleneck being the NS capabilities.

The requirements of packet forging attack are either an unauthenticated and unencrypted connection between the gateway and the NS (which is the case with many gateways using Semtech default Packet Forwarder), or for the attacker to own a malicious gateway and connect it to the victim's NS (which most of the time is easily achievable as most NSes offer guidelines on how to connect the gateway, and also offer roaming capabilities). The impact of such an attack is that the attacker can forge any packet and force the NS to accept it and forward it to the Application Server (AS). Since the Application Session Key is still unknown to the attacker, the decrypted payload of the modified packets seen in the AS will most likely be gibberish. Nonetheless, the attack will DoS the application, if the sequence number is set high enough. LoRaWAN 1.0.x defines the maximum gap between Frame Counters as 16384. This means that the attacker can only DoS 16384 LoRaWAN packets at a time, and would require to keep bruteforcing the MIC of frames with higher Frame Counters in order to maintain the DoS. LoRaWAN 1.1 removes this limit [13], which allows the attacker to forge a packet with the largest Frame Counter (2^{16} - 1 = 65535), increasing the length of the DoS. Bruteforcing the MIC might be even easier with the use of 32 bits Frame Counters (optional in LoRaWAN 1.0.x and default in LoRaWAN 1.1), as "since the FCnt field carries only the least-significant 16 bits of the 32-bits frame counter, the server must infer the 16 most-significant bits of the frame counter from the observation of the traffic." [13] For example, if the 32-bits Frame Counter of a LoRaWAN session reaches its 16-bits Frame Counter (FCnt) limit (65535), the next transmitted frame will have a FCnt=0. However, since the actual Frame Counter used is 32 bits, the Frame Counter will be equal to 65536. If more than 65536 frames transmitted by an end-node do not reach a gateway (e.g. because of jamming), the actual Frame Counter would have changed, although it might appear to the NS that the packets are in order. Depending on NS implementation, it can calculate the MIC of the received packet in one of the following ways: 1) using all permutations of the 16 Most-Significant Bits (MSB) of the Frame Counter until one of the calculated MIC values matches the MIC of the received packet; 2) based on the 16 MSB of the Frame Counter of the previously received valid packet.

If (2) is successful, in the case presented above (i.e. a gap of 65536 transmissions between reception of packets), a LoRaWAN connection will be permanently DoS-ed. If (1) happens, this effusively reduces the safety of a 32-bit MIC to only 16 bits, dramatically increasing the ease to conduct a packet forging attack.

PoC attack: The purpose of this attack was to test whether a LoRaWAN packet with the same FCntUp field (FCnt for uplink messages) but different Frame Counters (and thus different MICs) would be accepted by the NS. If this was the case, it would mean that any LoRaWAN packet could be forged in around 65535 tries. If this was not the case, it would mean that a DoS would happen naturally if a node was out of coverage for more than 65535 packets.

The gateway used was Kerlink Wirnet iFemtocell and Microchip RN2483 PICtail Daughter Board was used to send the LoRaWAN radio messages. The Things Network (TTN) was chosen as the NS since it supports 32-bit Frame Counters, used by our end-device. In order to implement the attack, we developed a Python application that extracts the LoRaWAN frame from a packet sent using Semtech Packet Forwarder (SPF) and which implements parts of the LoRaWAN 1.0.x specification in order to calculate all the possible MICs of a LoRaWAN frame. Fig. 2 shows the activity diagram of the

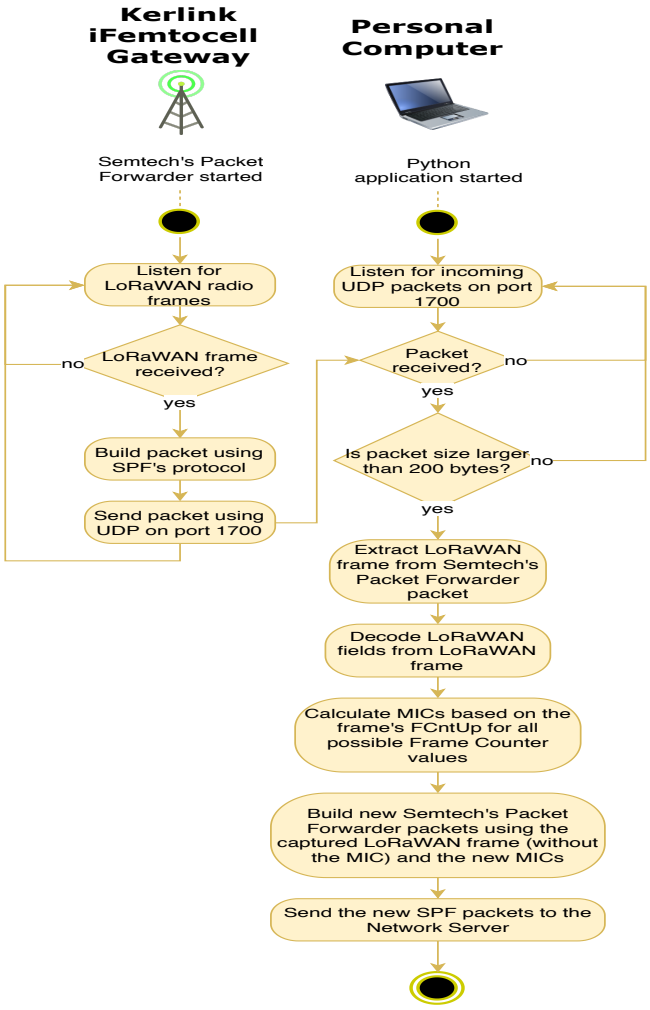

Fig. 2. Activity diagram showing MIC bruteforce attack workflow

attack. A LoRaWAN packet sent by the PICtail is received by the gateway, which forwards it using SPF to the PC (using UDP on port 1700) running the Python application. The application then extracts the LoRaWAN payload from the SPF packet (which are just base-64 encoded raw bytes) and decodes all its LoRaWAN fields including the 16-bit FCntUp. All the fields except the MIC are kept in the application. The new MICs are computed over those LoRaWAN fields, for increasing values of the Frame Counter. This is possible because the FCntUp represents only the 16 Least- Significant-Bits (LSB) of the Frame Counter. The rest 16 MSB of the Frame Counter are chosen from 0x0000 to 0xFFFF (2^{16} = 65536 possibilities). Once the MICs are computed, the application builds packets using the same protocol specified by SPF, and sends them to TTN NS. Not all 65536 packets are sent, since this may flood the server. It is enough to forward the first few packets (since they are built using ascending Frame Counters) to show whether the NS accepts these types of packets.

Fig. 3 shows the effect of the attack. The LoRaWAN packet that had its MIC bruteforced had FCnt=19 and a payload

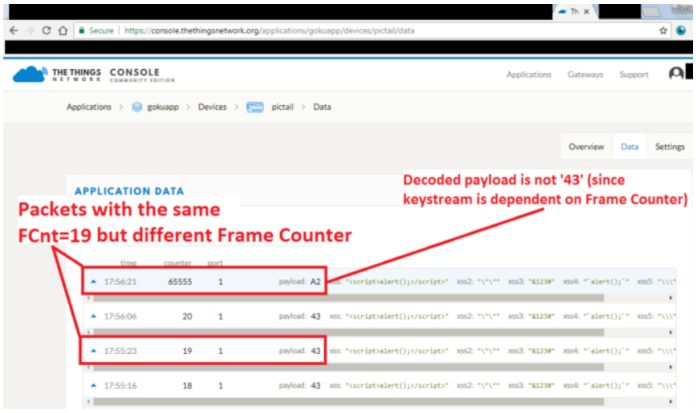

Fig. 3. The results of LoRaWAN packet forging attack

of 43 in hex ('C' in ASCII). The PICtail sent one more frame (FCnt=20) since otherwise the NS might discard some of the newly formed packets since they all have FCnt=19 and might resemble retransmissions. 10 packets were sent with FCnt=19 and the same LoRaWAN fields except for the MIC, which was different for increasing values of the Frame Counter (i.e. 19, 19+65536=65555, 65555+65536=131091, 131091+65536=196627, etc.).

As can be seen, after the packet with FCnt=20 was received by the NS, only the packet with the MIC for Frame Counter=65555 was accepted, so this bruteforce technique was not possible there. However, this also means that any device that is out of coverage for more than 65536 packets will be DoS-ed until manual intervention by the owner. Also shown in Fig. 3 is that the payload was decrypted differently, although it was originally sent as hex 43. This is because the keystream used to encrypt/decrypt the payload is also dependent on the Frame Counter, and in this case the payload was decrypted as A2. This shows that even if an attacker would be able to bruteforce the MIC of a frame, the payload decoded by the AS would be gibberish. This PoC also shows that an insecure Gateway-to-NS connection can be exploited by a malicious Man-In-The-Middle, who can then e.g. replay or selectively forward packets without the need to use more complex equipment such as an Software-Defined Radio (SDR).

B. Sigfox replay attack (with DoS of end-device)

To protect against replay attacks, Sigfox uses a 12-bit Sequence Number (SN) that is transmitted with every uplink frame and protected by Message Authentication Code (MAC). A Sigfox frame that has a lower SN than the latest received frame will be discarded by the Backend Server. The actual algorithm used to calculate the MAC is not public, but it uses AES in CMAC mode similar to LoRaWAN, with the secret NAK and the 12-bit SN (for uplink messages) as some of its inputs. For downlink messages, there is no public information related to the size of the SN, so it is impossible to say if they are more or less secure than uplink messages.

For uplink messages, a 12-bit SN only allows for 2^{12} = 4096 unique messages before overflowing back to 0. Combined

with the fact that Sigfox NAK key (used to calculate the MAC) does not change during the device lifetime, this leads to replay attacks being a highly likely threat to Sigfox. Maximum number of allowed UL messages for the most expensive Sigfox subscription (Platinum) is 140 per day, so sending that amount of messages one may cause the SN reset in 30 days. However, the maximum allowed UL messages is only a consequence of the Sigfox transmission time and the 1% duty-cycle limit. In practice, a Sigfox node can transmit messages well above the limit and still be accepted by the Sigfox Backend (albeit on a best-effort basis), decreasing the amount of time it would take to reset the SN. After the reset, the attacker can replay any of the previous 4096 packets indefinitely, as the security key NAK used to calculate the MACs is never changing, meaning the MACs will always be valid throughout the lifetime of the victim device. There is a caveat though, but one that may be bypassed: there is a maximum allowed gap between the SNs of consecutive Sigfox Frames before packets will be dropped. This gap depends on the subscription level and can be calculated as follows: Max(daily_max_transmissions * 3, 20), which translates to the gap being either 20, or 3 times the number of maximum allowed transmissions of the subscription level, whichever is higher. For Platinum subscription the maximum gap is 420.

If the attacker wants to DoS the Platinum-subscribed device for longest time possible by reaching SN = 4096, it is enough to replay 10 packets, each with its SN bigger than the SN of the victim by the maximum allowed SN gap (i.e. 420).

Another consequence of Sigfox SN gap limit is that end-nodes can be DoS-ed naturally, without the attacker intervening, if they run out of coverage for a prolonged period of time. The attacker can nevertheless make sure the device is DoS-ed by jamming an amount of packets equal to the maximum gap.

PoC attack: In order to observe the effect of message replay on Sigfox service, a controlled setup has been used, in which the 12-bit SN was reset back to 0. This required sending at least 2^{12} = 4096 Sigfox frames. A Sigfox end-device with a Platinum subscription was programmed to transmit a message every few seconds. A Faraday cage has been used to prevent most of the Sigfox messages from radiating outside and getting to the Sigfox Gateway (as per the duty-cycle requirements). The Sigfox device was periodically taken out of the Faraday cage in order to allow for some messages to get to the gateway (so the current SN could be checked at the Sigfox Backend). A few Sigfox frames were intercepted in the beginning of the experiment, when the SN was low ($\approx$ 200), using a SDR device. Once enough frames were sent by the end-device so that the SN would reset, the captured frame was replayed, while checking the Sigfox Backend to see the effects of the attack. The Sigfox end-device used was a Thinxtra Xkit development board, while the device intercepting and replaying the Sigfox frames was HackRF One.

The effects of this replay attack can be seen in Fig. 4. The captured Sigfox packet had a SN of 258. Once the Thinxstra Xkits SN reset back to 0 (after a few hours), and after $\approx$ 90 more transmissions the previously captured packet

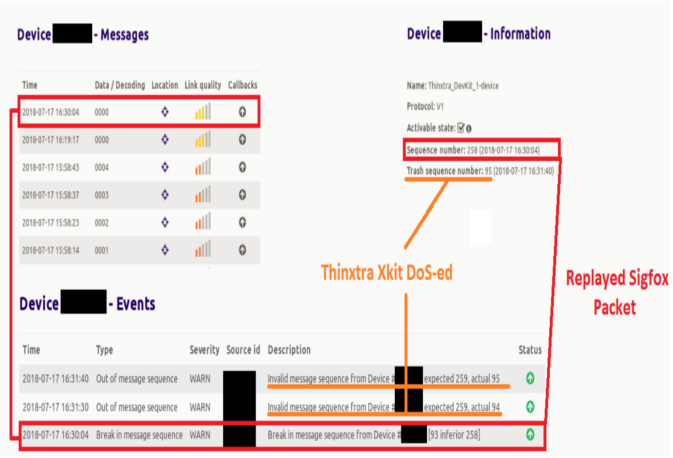

Fig. 4. The results of Sigfox attack

was replayed. The chronological events shown in Fig. 4 were the following: 1) The Thinxtra Xkit SN reaches 93; 2) The previously captured packet with SN 258 is replayed and it successfully reaches the Backend at 16:30:04 and is displayed in the console (payload 0000). A warning is raised in the Backend that says that packets have been lost (since SN 258 arrives after SN 93, meaning that 258 - 93 = 165 were theoretically lost); 3) Thinxtra Xkit continues to send frames with SN 94 and 95, but they are dropped by the Backend.

As shown in this PoC, Sigfox is vulnerable to replay attacks due to the small size of its SN (only 12 bits), and this can lead to an attacker injecting previously sent messages into the system, as well as DoS-ing the end-device.

C. NB-IoT

Scan using malicious UE: As UEs can send IP data, classical attacking techniques over IP (e.g. port scanning, ARP spoofing, DNS spoofing) may be possible inside the NB-IoT network as well. In order to receive an IP packet from the mobile network, a UE needs to first set up a Packet Data Protocol context by selecting an Access Point Name (APN). This connects the UE to a Packet Data Network Gateway (PGW) and grants it access to a private network. An enterprise that uses a large number of UEs will likely have them connected to the same private network (LAN). This means that if an attacker gets access to an UE (e.g. by stealing it), they may use it to attack other UEs which are part of the same private network. For example, if the victim's UE opens a port to communicate with an IoT Server, the attacker using a malicious UE could send malicious data to the open port, masquerading as the IoT Server. Another impact this attack could have is DoS by battery exhaustion. For example, an attacker might force a UE to receive and send data (e.g. by pinging the victim's UE), draining the victim battery, which would lead to a permanent DoS (forcing the device owner to replace the UE battery).

The network diagram of the setup can be seen in Fig. 5 and consists of an Android-based LTE UE (as shown in [5],

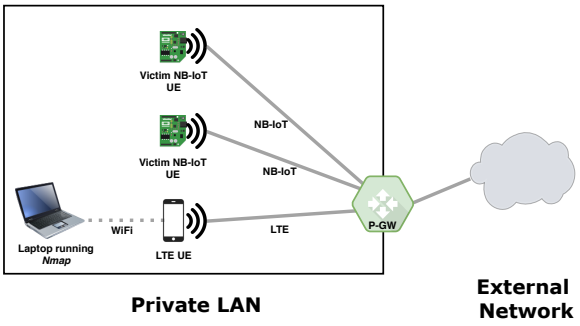

Fig. 5. NB-IoT scan using malicious UE

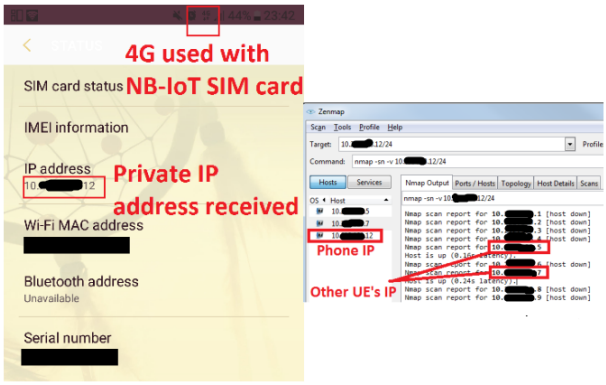

Fig. 6. NB-IoT scan results

NB-IoT SIM card can be misused in 4G) that is connected to a European mobile operator's NB-IoT APN using a SIM card for connecting an NB-IoT UE to that same network. The same LTE UE was used to create a WiFi hotspot in order to allow a PC to join the private APN network. The PC can be then the source of the attack, allowing any PC tools to be used for it.

Due to the fact that the setup included a real network, no malicious activity was performed, and only a ping scan/sweep was performed on the private LAN to showcase that this attack is indeed possible. A ping scan sends ICMP requests to every IP on a network in order to detect which devices are online. For this, a tool called Zenmap was used.

Fig. 6 shows the results of the scan. The IP address received by the LTE UE was 10.X.X.5, which is a private IP, meaning that the private network is behind a router that does Network Address Translation (NAT). Based on this IP, the subnet made of IPs 10.X.X.0-255 was scanned using the nmap command *nmap -sn 10.X.X.12/24*. Apart from the phone's IP, two other NB-IoT devices were found, and their IP addresses identified. At this point, an attacker could scan for open UDP and TCP ports on the found NB-IoT devices and then send forged messages to those open ports.

IV. Conclusion

In this paper we presented a vulnerability analysis of LP-WAN technologies. We provided generic considerations on security threats towards any LP-WAN standard, and we described 3 attack scenarios, summarised in Table I. The tests

TABLE I
Summary of LPWAN attacks

Target LPWAN	LoRaWAN	Sigfox	NB-IoT
Threat	Packet forging	Replay	Malicious UE
Impact	Message injection (gibberish payload), DoS (MIC bruteforce)	Message injection, DoS	Private NB-IoT network infiltrated
Cause	Short (4-byte) MIC	12-bit SN, max. SN gap	Poor protection from UEs private network
Exploitability	Easy (1.0.x) / Hard (1.1)	Easy	Medium/Hard
Prevalence	Common (1.0.x) / Rare (1.1)	Common	Common

show that Sigfox in its current state should not be used for critical applications (unless better replay protection is built at a higher layer by end users, which would decrease the already small payload size). LoRaWAN and NB-IoT offer sufficient security guarantees, but they need to be properly enforced. For LoRaWAN, we show that packets could be forged in some situations, forcing the AS to receive a packet with garbage payload and possibly temporarily causing a DoS. Thus, LoRaWAN applications must be "garbage-proof" and the AS/NS should disallow the communication from the devices sending invalid packets to prevent DoS. For NB-IoT, before deploying critical applications the user should ensure that their operator enforces best security practices on the network.

References

[1] "Sigfox technology," accessed: 11-02-2019. [Online]. Available: https://www.sigfox.com/en
[2] "Lorawan technology," accessed: 11-02-2019. [Online]. Available: https://www.lora-alliance. org/
[3] "Nb-iot technology," accessed: 11-02-2019. [Online]. Available: https://www.gsma.com/iot/ narrow-band-internet-of-things-nb-iot/
[4] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "Ddos-capable iot malwares: Comparative analysis and mirai investigation," 2018.
[5] T. Xie, C.-Y. Li, J. Tang, and G.-H. Tu, "How voice service threatens cellular-connected iot devices in the operational 4g lte networks," *2018 Ieee International Conference on Communications (icc)*, vol. 2018-May, pp. 6 pp., 6 pp., 2018.
[6] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in lorawan," *Proceedings - Acm/ieee International Conference on Internet of Things Design and Implementation, Iotdi 2018*, pp. 129–140, 2018.
[7] I. Butun, N. Pereira, and M. Gidlund, "Analysis of lorawan v1.1 security," *Proceedings of the 4th Acm Mobihoc Workshop*, pp. 1–6, 2018.
[8] E. Kail, A. Banati, E. Laszlo, and M. Kozlovszky, "Security survey of dedicated iot networks in the unlicensed ism bands," *2018 Ieee 12th International Symposium on Applied Computational Intelligence and Informatics (saci)*, pp. 449–453, 2018.
[9] S. Chacko and M. D. Job, "Security mechanisms and vulnerabilities in lpwan," *Iop Conference Series: Materials Science and Engineering*, vol. 396, no. 1, p. 012027, 2018.
[10] J. Liu, Y. Yu, F. X. Standaert, Z. Guo, D. Gu, W. Sun, Y. Ge, and X. Xie, "Small tweaks do not help: Differential power analysis of milenage implementations in 3g/4g usim cards," *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9326, pp. 468–480, 2015.
[11] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "Iot goes nuclear: Creating a zigbee chain reaction," *Ieee Security and Privacy*, vol. 16, no. 1, pp. 54–62, 2018.
[12] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, P. D. Gallagher, and U. Secretary For, "Nist special publication 800-57 recommendation for key management part 1: General," 2013.
[13] L. Alliance, "Lorawan 1.1 specification," 2017, accessed: 18-02-2019. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf