Hit the KeyJack: stealing data from your daily wireless devices incognito

Guillaume Fournier, Pierre Matoussowsky, Pascal Cotret

Hit the KeyJack: stealing data from your daily wireless devices incognito

Guillaume Fournier[1], Pierre Matoussowsky[1], Pascal Cotret[1,2]

[1] CentraleSupélec - Rennes, France
guillaume.fournier@supelec.fr, pierre.matoussowsky@supelec.fr
[2] IETR, SCEE Research Team - Rennes, France
pascal.cotret@centralesupelec.fr / @Pascal_r2

Abstract. Internet of Things (IoT) is one of the most fast-growing field in high technologies nowadays. Therefore, lots of electronic devices include wireless connections with several communication protocols (WiFi, ZigBee, Sigfox, LoRa and so on). Nevertheless, designers of such components do not take care of security features most of the time while focusing on communication reliability (speed, throughput and low power consumption). As a consequence, several wireless IoT devices transmit data in plaintext creating lots of security breaches for both eavesdropping and data injection attacks. This work introduces KeyJack, a preliminary proof-of-concept of a solution aiming to eavesdrop wireless devices and hopefully perform injection attacks afterwards. KeyJack operates on widely-used devices: our keyboards! This solution is based on low-cost embedded electronics and gives an attacker or a white hat hacker the possibility to retrieve data from John Doe's computer. This work also shows that this approach could be used to any wireless device using 2.4GHz radio chips like the NRF24L01 from Nordic Semiconductor.

1 Introduction

Nowadays, most of the people have IoT devices at home or even at work: for instance, it could be included in a fridge, a set-top box or computers. IoT is a highly-growing field and it will get even bigger in the next decade (a Verizon report [21] foresees that the IoT market will hit the $1 trillion limit by 2019). In the same document, it can be seen that the IoT market targets several applications such as healthcare and home monitoring. Most of these connected devices were designed to perform tasks fast and in a cheap way (in other words, communication links have to be fast and low-power). Unfortunately, without security, each device is a vector of threats: malevolent people could use breaches in wireless protocols to steal or inject data using man-in-the-middle (MITM) attacks.

In this context, this work focuses on wireless keyboards that are widely-spread components. This work presents KeyJack that is a kind of wireless keylogger implemented in a tiny electronic board aiming to retrieve data from a remote computer. Furthermore, this work gives some clues about using it as a malicious injection device.

Section 2 presents some related works of both classic keyloggers and embedded electronics solutions with similar goals. Section 3 presents KeyJack, its requirements and implementation details. Then, Section 4 presents an eavesdropping scenario where Keyjack is used and gives some hints about the feasability of injection attacks using such hardware components. Finally, Section 5 presents some conclusions and perspectives about this work.

2 Related works

2.1 Keyloggers

When an hacker wants to retrieve data from a user keyboard, keyloggers could be used [18,19,12,5]: these components are usually softwares running in the background of the target computer. In more recent works such as Damopoulos et al. [4], keyloggers are also implemented for touchscreen that are widely-spread interfaces on our smartphones and tablets. Such keyloggers aim to keep a copy of each keyboard hit made by the victim. On the other side, several works such as [10,17] present countermeasures in order to implement systems resilient to such attacks. Therefore, IoT designers could imagine to create embedded systems immune to standard keylogger implementations.

However, most of software keyloggers are not so easy to use:

- Advanced features are rarely included in free versions. As keyloggers may be used for ”bad” purposes, developers keep the most intrusive options for paid licenses.
- Basic keyloggers are not 100% discrete as they appear in the task manager.
- For some of them, administrator rights may be required by the operating system (for instance, installing a driver).
- Furthermore, results take up space on the target computer and may increase its power consumption.

2.2 Other interesting works

In the context of pure keyloggers, there are other interesting alternatives in the hardware community:

- KeyGrabber USB made by KeeLog [20].
- USB Rubber Ducky, a USB tool by Hak5 [9]. This USB key includes a 60MHz programmable microcontroller and a μSD slot. It behaves like a keyboard: therefore, nearly anything can be performed (from a Rick Roll hack to keyloggers as well).

Both solutions look like USB flash drives: it can be easily hidden on a computer port. Even if some countermeasures exists (for instance, KeyScrambler[14] encrypts of all keyboard hits in Firefox), it is not 100% satisfying as it still needs some physical access to the target. Furthermore, even if such a tool may be hidden in the task manager, it is assumed that its power consumption may be revealed with physical measurements.

KeyJack wants to tackle those problems using an alternative breach: nowadays, most keyboards are wireless and may be affected by eavesdropping and MITM attacks. The next subsection presents some works using wireless connection to reveal security breaches.

2.3 Embedded electronics solutions

There are several works aiming to steal information from wireless computer devices. The most "industrial" solution is MouseJack from Bastille Networks[3]. MouseJack is an exploit used in several wireless (non-Bluetooth) keyboards that can be used to perform eavesdropping and relay attacks. However, a laptop is required to run MouseJack in contradiction to KeyJack which aims to be a discrete and standalone solution [11].

Other people tried to make things smaller. Digital Security (`@iotcert`) implemented related eavesdropping methods for Bluetooth devices running on a single-board computer such as Raspberry Pi [2,3]: Cauquil et al. implemented a Bluetooth sniffer on a RaspberryPi Zero single-board computer that can be used to track people and steal secrets (as it was shown in their Nuit du Hack'16 talk [4]).

Samy Kamkar (`@samykamkar`) developed Keysweeper [13,7], a solution based on a small tiny microcontroller similar to Teensy or Arduino Nano. Even if this solution is the most similar to KeyJack, it does not take into account the feasability of data injection (Kamkar only focused on data listening). Furthermore, KeyJack plans to use a GSM chip to perform multiple eavesdropping with target localization in a use case where several KeyJack devices would be implemented.

3 KeyJack

3.1 Threat model

There are several keyboard manufacturers. This work focuses on the two main ones: Microsoft and Logitech (it is assumed that generic/low-cost keyboards may work as well). Microsoft/Logitech keyboards uses a classic Wifi-based protocol working at 2.4GHz (for European versions at least). When this 2.4GHz link is left unencrypted, a classic MITM scheme could be used as shown in Figure 1.

[3] `https://www.mousejack.com/`

[4] `http://virtualabs.fr/ndh16/ndh16-mass-pwning-bug.pdf`

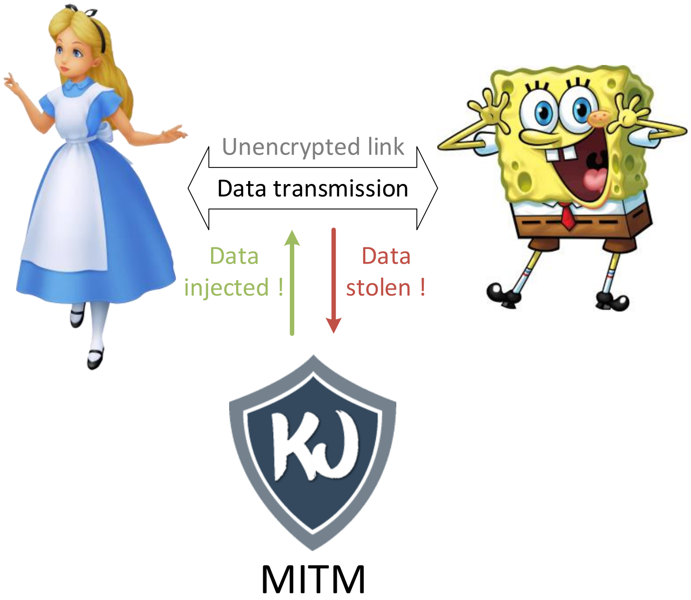


Fig. 1. MITM scheme with KeyJack as the eavesdropper/injection device

For some recent keyboards, the communication between Alice and Bob is encrypted: even if such schemes may be broken using brute force or other advanced techniques, this work assumes that the link is left in plaintext:

- For basic models, security was not implemented in the wireless protocol.
- For future perspectives, KeyJack could be adapted to other plaintext protocols working at different frequencies (5GHz band, for instance).

3.2 Requirements

A KeyJack device must have the following requirements:

- No physical access to the target device/computer. It means there will not be any USB connection or malicious software installed.
- Tiny implementation: in other words, KeyJack must be a small-sized board, easily transportable and autonomous in terms of energy.
- "Tracking-friendly": KeyJack end-users must be able to get eavesdropping results from a remote device (website, smartphone. . .).
- Injection-enabled: this work aims to propose a solution which enable not only eavesdropping but also data injection (from a remote interface as well).
- This work focuses on a Microsoft Wireless Keyboard 800[5]. Mainly because its protocol was unecrypted.

[5] https://www.microsoft.com/accessories/en-gb/products/keyboards/wireless-keyboard-800/2vj-00006

- For further implementations, a GSM chip in order to retrieve locations of KeyJack nodes in case we want to install a network of such devices.
- And, of course, something that is low-cost and easily reproductible!

3.3 Hardware implementation

KeyJack components are shown in Figure 2:

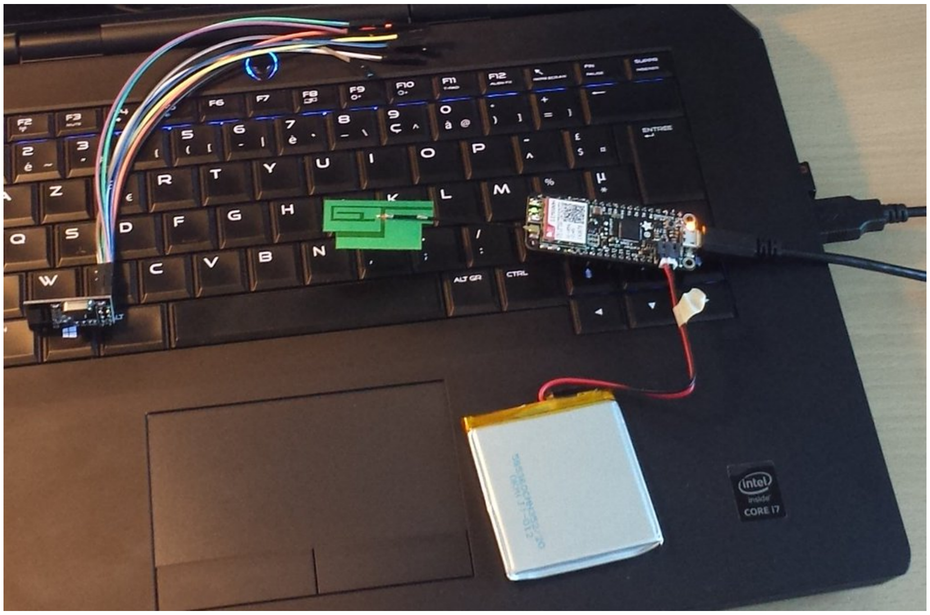


Fig. 2. Electronic components used in KeyJack

From the left to the right side:

- A 2.4GHz board based on a NRF24L01 chip from Nordic Semiconductors.
- An antenna.
- Adafruit Feather FONA as the microcontroller board: this board includes, in a tiny form factor, an ATmega32u4 running at 8MHz and a GSM chip.
- And a battery.
- (the USB cable is just here for programming purposes)

3.4 Software layer

Figure 3 shows an example of a KeyJack network. Each node is the Adafruit platform described in the previous section running a given Arduino code. When each node collects information, it is transmitted to the self-hosted server where an internal website was developed.

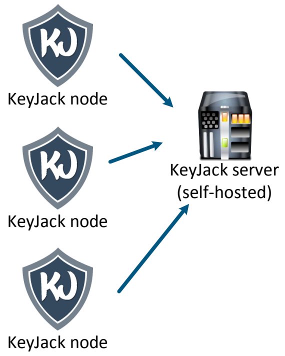
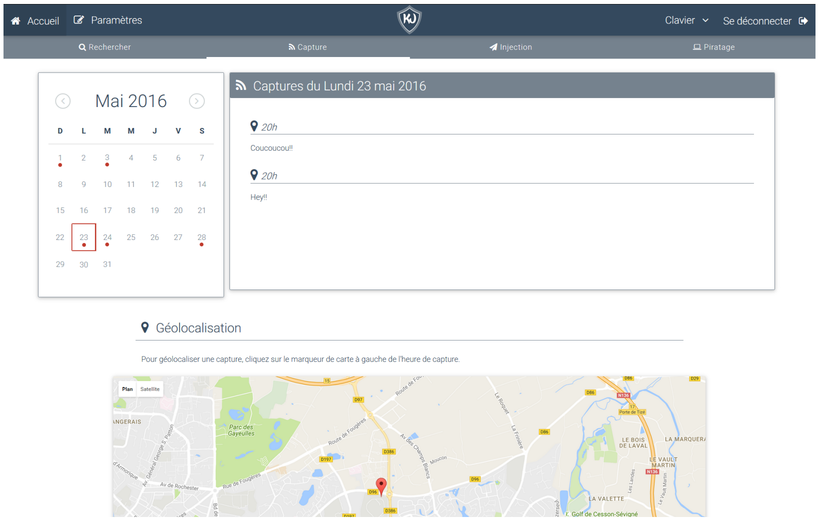


Fig. 3. Example of a KeyJack network with three nodes and a server

On the server side, KeyJack interface looks as follows:



Mai 2016

D	L	M	M	J	V	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Captures du Lundi 23 mai 2016

20h

Coucoucou!!

20h

Hey!!

Géolocalisation

Pour géolocaliser une capture, cliquez sur le marqueur de carte à gauche de l'heure de capture.

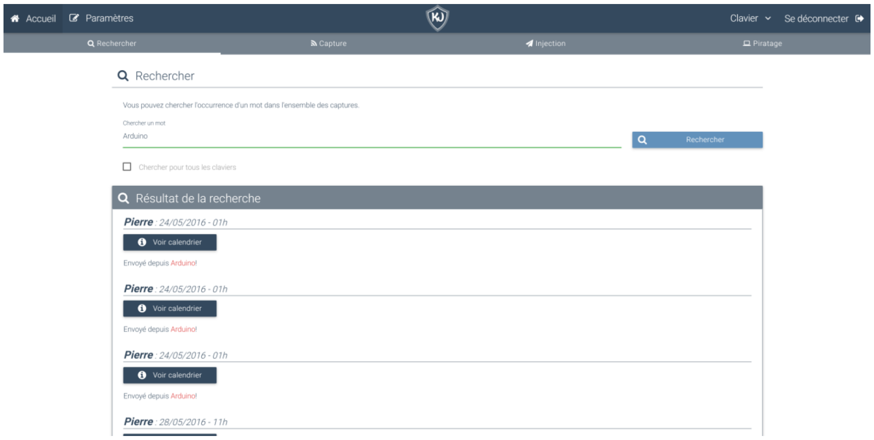
Fig. 4. KeyJack server interface

4 Case study: eavesdropping a Microsoft keyboard

As it was said in Section 3, this case study is focused on a Microsoft Wireless Keyboard 800. Furthermore, we only used a single KeyJack node as shown in Figure 2. When the user enters the KeyJack server, an interface as shown in Figure 6 appears.

Each keyboard is identified by its MAC address and has a dedicated menu:

- *Search*. In this part, the user can read logs of former measurements.
- *Capture*. Reading captures filtered by their date.
- *Injection*. Transmitting keyboard keys.
- *Hacking*. This last tab allows to launch attack scripts.



Vous pouvez chercher l'occurrence d'un mot dans l'ensemble des captures.

Chercher un mot: Arduino

□ Chercher pour tous les claviers

Pierre : *24/05/2016 - 01h*

Envoyé depuis Arduino!

Pierre : *24/05/2016 - 01h*

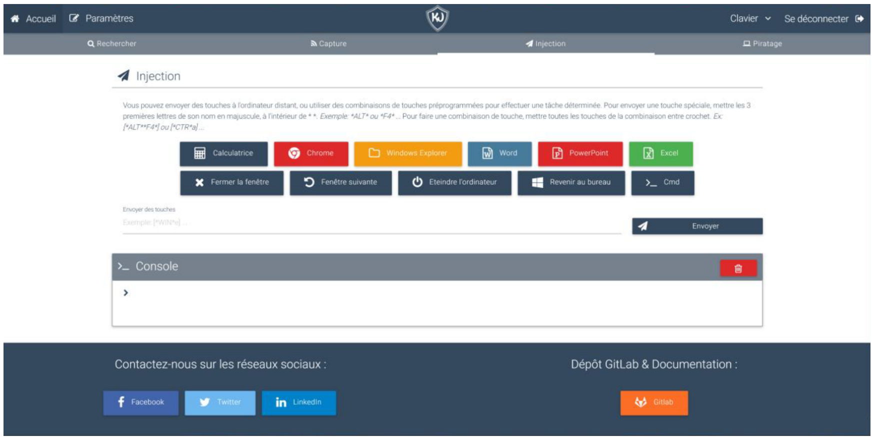
Envoyé depuis Arduino!

Pierre : *24/05/2016 - 01h*

Envoyé depuis Arduino!

Pierre : *28/05/2016 - 11h*

Fig. 5. KeyJack keyboard interface

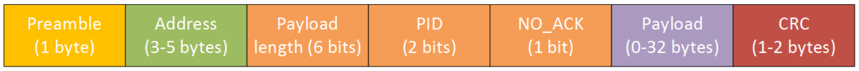


Vous pouvez envoyer des touches à l'ordinateur distant, ou utiliser des combinaisons de touches préprogrammées pour effectuer une tâche déterminée. Pour envoyer une touche spéciale, mettre les 3 premières lettres de son nom en majuscule, à l'intérieur de * *. *Exemple: *ALT* ou *F4** ... Pour faire une combinaison de touche, mettre toutes les touches de la combinaison entre crochet. *Ex: [*ALT**F4*] ou [*CTR*a]* ...

Envoyer des touches: Exemple: [*WIN*e] ...

Fig. 6. KeyJack injection interface

Normally, the NRF24L01 chip is not able to work as a sniffer: in fact, the target MAC address is needed and it is not possible to scan the frequency spectrum around 2.4GHz to find one. However, as Samy Kamkar explained in its Keysweeper project, we can send fake information about the MAC address in order to swindle the wireless chip.

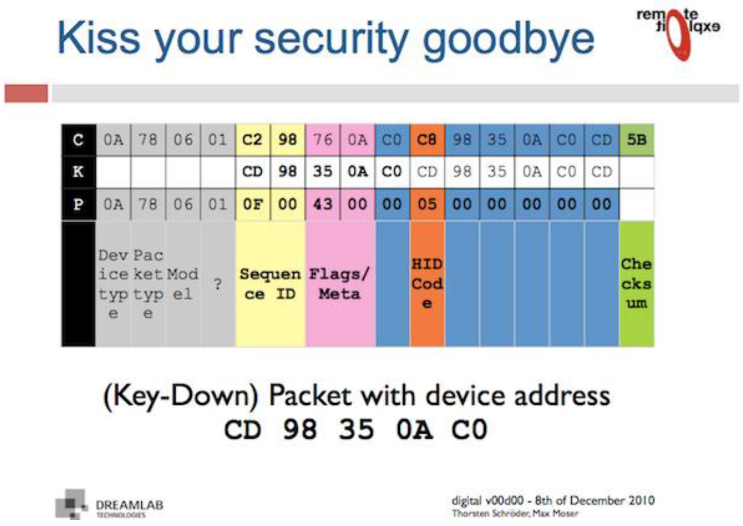


Preamble (1 byte)	Address (3-5 bytes)	Payload length (6 bits)	PID (2 bits)	NO_ACK (1 bit)	Payload (0-32 bytes)	CRC (1-2 bytes)

Fig. 7. NRF24L01 packet structure

With previous works of Travis Goodspeed, Samy Kamkar discovered that if we enter an incorrect value regarding the MAC address size (writing the preamble itself), the upset NRF24L01 chip considers all preambles as the target MAC address. However, all data after the MAC address should be the payload. Therfore, we get all traffic packets with the MAC address and everything else up to the CRC. From there, we can proceed to keyboard detection.

Each brand has its own protocol to deal with the USB dongle on the computer. Microsoft does not encrypt data sent by its keyboards (it is only done since 2015 !). The only security measure is a XOR performed on the payload with the MAC address. As we know how to get MAC address, this is not a problem for us. As a consequence, we only have to detect Microsoft protocols and perform a XOR.



Kiss your security goodbye

C	0A	78	06	01	C2	98	76	0A	C0	C8	98	35	0A	C0	CD	5B
K					CD	98	35	0A	C0	CD	98	35	0A	C0	CD	
P	0A	78	06	01	0F	00	43	00	00	05	00	00	00	00	00	
	Device type	Packet type	Model	?	Sequence ID		Flags/ Meta			HID Code						Checksum

(Key-Down) Packet with device address
CD 98 35 0A C0

digital v00d00 - 8th of December 2010
Thorsten Schröder, Max Moser

Fig. 8. Packet structure

Samy Kamkar also discovered that all Microsoft keyboards used a MAC address beginning with `0xCD` (this is the only byte we need in further measurements). In fact, Microsoft is done in a way that the key value is in the 10^{th} position. As the MAC address is 5 bytes wide and 4 first bytes are not encrypted, the key value is always XORed with `0xCD`.

The last criteria we use to detect Microsoft keyboards is the value of the first unencrypted bytes. As it is shown in Figure 8, the device type is always `0x0A` (for keyboards). Then, the packet type indicates the key category (we only focus on keystrokes or idles). Related codes are `0x78` and `0x38`. We are finally ready to scan, for all frequencies from 2403MHz to 2480MHz:

- We check if the MAC address of the transmitter begins by `0xCD`.
- We test if the paylod begins by `0xA78` or `0xA38`.

5 Conclusion and perspectives

This work presented KeyJack, a low-cost solution for basic eavesdropping of a specific Microsoft wireless keyboard. The proof-of-concept is a standalone device which can be left in any open-space and small enough to be hidden from people sight. Even if this study focuses on a specific keyboard model, there are opportunities with other models/vendors when communications are not encrypted. Furthermore, KeyJack can be easily modified for data injection as the server side is already implemented. As each keyboard is clearly identified, the next perspective is to make a network of KeyJack node and a single server where we could monitor everything from a remote location. Finally, KeyJack may be adapted for other protocol where security matters in the context of Internet of Things.

References

1. Bastille Networks, L.: Mousejack faq (Feb 2016), `https://www.bastille.net/sites/all/themes/professional_theme/media/bastille_mousejack_faq_2_12_16-2.pdf`
2. Cauquil, D.: Bluetooth low energy device spoofing library (Aug 2016), `https://github.com/DigitalSecurity/mockle`
3. Cauquil, D.: Btlejuice: The bluetooth smart mitm framework (Aug 2016), `https://speakerdeck.com/virtualabs/btlejuice-the-bluetooth-smart-mitm-framework`
4. Damopoulos, D., Kambourakis, G., Gritzalis, S.: From keyloggers to touchloggers: Take the rough with the smooth. Computers & Security 32, 102 – 114 (2013), `http://www.sciencedirect.com/science/article/pii/S0167404812001654`
5. Dorne, M.K.: La menace keylogger (Jul 2016), `http://korben.info/la-menace-keylogger.html`
6. Express, J.: Vous utilisez des objets connectés? gare à vos données (Jul 2016), `http://lexpansion.lexpress.fr/high-tech/vous-utilisez-des-objets-connectes-gare-a-vos-donnees_1813593.html`

7. Goodin, D.: Meet keysweeper, the 10usd usb charger that steals ms keyboard strokes (Jan 2015), http://arstechnica.com/security/2015/01/meet-keysweeper-the-10-usb-charger-that-steals-ms-keyboard-strokes/
8. Hacker, B.: La sécurité dans l'internet des objets (Jul 2016), http://www.leblogduhacker.fr/la-securite-dans-internet-des-objets/
9. Hak5: Usb rubber ducky wiki (Jan 2016), http://usbrubberducky.com/#!index.md
10. Herley, C., Florencio, D.: How to Login from an Internet Cafe Without Worrying about Keyloggers . Association for Computing Machinery, Inc (2003)
11. Higgins, K.: Mousejack attack bites non-bluetooth wireless mice (Feb 2016), http://www.darkreading.com/endpoint/mousejack-attack-bites-non-bluetooth-wireless-mice/d/d-id/1324404
12. Holz, T., Engelberth, M., Freiling, F.: Learning More about the Underground Economy: A Case-Study of Keyloggers and Dropzones, pp. 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04444-1_1
13. Kamkar, S.: Keysweeper (2015), http://samy.pl/keysweeper/
14. KeyScrambler: Keyscrambler's features (Jan 2016), https://www.qfxsoftware.com/ks-windows/features.htm
15. Lifchitz, R.: Zigbee security review of a famous french set-top box (Jul 2016), https://speakerdeck.com/rlifchitz/zigbee-security-review-of-a-famous-french-set-top-box
16. Moine, F.: Huit bonnes pratiques pour bien sécuriser les objets connectés (Jul 2016), http://www.journaldunet.com/solutions/expert/64817/huit-bonnes-pratiques-pour-bien-securiser-les-objets-connectes.shtml
17. Ortolani, S., Crispo, B.: Noisykey: Tolerating keyloggers via keystrokes hiding. In: Presented as part of the 7th USENIX Workshop on Hot Topics in Security. USENIX, Berkeley, CA (2012), https://www.usenix.org/conference/hotsec12/workshop-program/presentation/Ortolani
18. Sagiroglu, S., Canbek, G.: Keyloggers. IEEE Technology and Society Magazine 28(3), 10–17 (Fall 2009)
19. Subramayam, K., Frank, C., Galli, D.: Keyloggers: The Overlooked Threat to Computer Security. The Vespiary (2003)
20. USB, K.: Hardware keylogger - keygrabber usb (Jan 2016), https://www.keelog.com/usb_hardware_keylogger.html
21. Verizon: State of the market: The internet of things 2016 (Jan 2016), https://www.verizon.com/about/sites/default/files/state-of-the-internet-of-things-market-report-2016.pdf