Analysing the Resilience of the Internet of Things against Physical and Proximity Attacks

He Xu[1,2], Daniele Sgandurra[3(✉)], Keith Mayes[3], Peng Li[1,2], and Ruchuan Wang[1,2]

[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
xuhe, lipeng, wangrc@njupt.edu.cn,
[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China
[3] Information Security Group, Royal Holloway, University of London, Surrey TW200EX, UK
daniele.sgandurra, keith.mayes@rhul.ac.uk

Abstract. The Internet of Things (IoT) technology is being widely integrated in many areas like smart-homes, smart-cities, healthcare, and critical infrastructures. As shown by some recent incidents, like the Mirai and BrickerBot botnets, security is a key issue for current and future IoT systems. In this paper, we examine the security of different categories of IoT devices to understand their resilience under different security conditions for attackers. In particular, we analyse IoT robustness against attacks performed under two threat models, namely (i) physical access of the attacker, (ii) close proximity of the attacker (i.e., RFID and WiFi ranges). We discuss the results of the tests we performed on different categories of IoT devices, namely IP cameras, OFo bike locks, RFID-based smart-locks, and smart-home WiFi routers. The results show that most of IoT devices do not address basic vulnerabilities, which can be exploitable under different threat models.

Keywords: IoT, Smart Home, IoT Attacks, Threat Models

1 Introduction

The Internet of Things (IoT) is the interconnection of billions of "smart" devices to the Internet, from smart-lights, smart-door locks, smart-air conditioners, smart-cameras, to intelligent fridges, and even vehicles. These objects are typically networked devices that bridge the physical and virtual worlds. The growth of IoT in recent year is significant, as consumers, businesses, and governments recognize the benefit of interconnecting these devices together to provide additional features. By 2020, it is estimated that 24 billion IoT devices will be installed world-wide [2]. Furthermore, IoT market is set to increase from $1.9 trillion in 2013 to $7.1 trillion by 2020 [1]. However, many IoT devices are manufactured with minimal security considerations, which render them an

easily-exploitable target for attackers. As an example, on October 21st of 2016, the Mirai IoT botnet [16,17], a network composed by hundred of thousands of IoT devices controlled by an attacker, launched a distributed denial-of-service (DDoS) attack against DYN, a major DNS provider. The attack generated 1.2 terabits of malicious traffic forcing DYN off the Internet for hours. This botnet was largely formed by vulnerable IP cameras, digital video recorders, and routers. Some months later, an allegedly white hacker, named Janit0r, claims to have "bricked" more than 2 million IoT devices since January 2017 in an attempt to "protect" the devices before they could be enslaved by Mirai [4] [3]. While the initial motivations of this attack (called Brickerbot) were to "teach" a lesson to the IoT industry to improve the security of IoT devices, it has actually achieved to permanently create a DoS on these devices. All of these attacks are just some examples of the security risks faced by current IoT devices, which are mainly due to a lack of proper security testing by several IoT manufacturers. The goal of this paper is to analyse the attack surface of some notable categories of IoT devices to understand their resiliency under different attack scenarios.

The rest of the paper is organised as follows. In Sect. 2 we will review some notables related works on IoT attacks. In Sect. 3, we examine the security of some representative classes of IoT devices and describe some possible attacks under two threat models: physical attack and proximity attack. Section 4 reports the details of our attacks under these threat models, which show that IoT devices contains vulnerabilities that can be easily exploited using different attackers' capabilities. Finally, Sect. 5 concludes the paper.

2 Related Works

In [11], the authors have analysed the security of Samsung's SmartThings platform and found several security vulnerabilities. Similarly, Eyal Ronen and Adi Shamir [19] have performed functionality extension attacks on IoT devices using smart-lights as a covert LIFI communication system to exfiltrate data from a highly secure office building. The authors were able to read the leaked data from a distance of over 100 meters using cheap equipment. The authors of [18] have designed a feature-distributed malware to perform various malicious activities, such as unlocking smart-locks and disarming security alarms. These results show that traditional web attack techniques, such as cookie stealing, can be turned into sophisticated attacks on IoT devices. Similarly, the authors of [13] examine the security of five commercial home smart-locks, and show that most of these devices suffer from poor design and implementation choices. In addition, the authors of [21] use an existing Apple app (called Loki), which is a survey app that integrates the authors' malware and approved by Apple Store, to infiltrate home networks. The results show that home routers are poorly protected against some attacks. Bertino et al. analyse the IoT vulnerabilities [9] from insecure web/mobile/cloud interface, insufficient authentication/authorization, insecure network services, lack of transport encryption/integrity verification, privacy concerns, insufficient security configurability, insecure software/firmware, and poor

physical security. However, none of these papers has analysed the risk of IoT vulnerabilities by including a detailed threat model, which instead is a required condition (i) to understand the privileges needed by an attacker to perform an attack [20], (ii) to enable manufacturers to address security methodically by using different security assumptions. This is the main goal of this paper.

3 Physical and Proximity Attacks for IoT

We describe two existing main threat models, which should be considered when designing and testing IoT devices for security. To illustrate these threat models, we will refer to Device-Cloud-Mobile (DCM) model shown in Figure 1, as it is widely used today in commercial products. Here, the IoT devices (IP camera, smart-lock, and bike lock) need to connect to the back-end Cloud system, and support services by an user app running on mobile devices. In some case, the app connects directly with the IoT device using the device Wi-Fi hotspot capabilities, which either processes the requests directly or bridges it to the Cloud backend. We will refer to the legitimate owner of IoT devices as "Alice", and we will user the generic term "attacker" to refer to a generic attacker.

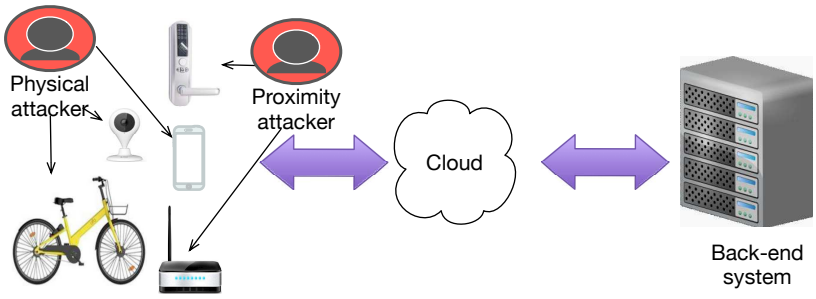

Fig. 1. Device-Cloud-Mobile IoT Architectural Model

Physical Attack. In this threat model, an attacker can physically interact with the IoT device without notification to Alice. For example, the attacker can access the hardware, and read and modify the default IoT device's settings, which may impact the privacy and authentication credentials of Alice. An example of such attacks is when an IoT device is left unattended for short or long period of time, or when it is publicly available (e.g., a sensor in an open field), which enables an attacker to achieve unguarded access to it.

Proximity Attack. Several IoT devices use wireless communication technology, such as Bluetooth, infrared, ZigBee, and RFID to communicate data to a central station or with other devices (e.g., a smart-phone or another node). Therefore,

many of theses communications, if not properly protected, might be available for listening or modification in proximity areas. In this attack, an attacker can observe Alice's interactions with the IoT devices and can also interact with them by injecting data into the available channel. However, this type of attacker does not possess an authorized device and cannot physically alter its setting or firmware. For example, through the analysis of the wireless communication, an attacker can perform replay or clone attacks. In addition, several WiFi-based IoT devices can become a client of a botnet [9] because of their vulnerable OS, such as a vulnerable BusyBox-based Linux distribution (the preferred target of several IoT botnets). Therefore, the WiFi-based IoT device can be used to hide the botnet software, which may be exploited to remotely control the IoT device. This type of attacker does not possess the authorized device and cannot physically alter it. The two classes of adversaries are also shown in Fig. 1.

4 Performing Physical and Proximity Attacks on IoT

In the following, we describe eight attacks we have analysed and tested under the previous threat models. In detail, they are: physical attacks for IP camera and OFo bike lock, proximity attacks for RFID-based smart-locks, and proximity attacks for WiFi routers in smart-home environment. Table 1 details each IoT device we studied, their architecture, the analysed threat model and the attack we have performed.

Table 1. Details of the Tested IoT devices

Type of devices	Device	Architecture	Threat Model	Attacks
IP Camera	360 Camera	DCM	Physical Attack	Default Login Reset password
Smart Lock	OFo Bike Lock	No Direct Internet Connection	Physical Attack	Default Login Reset Password Malicious QR Code
Smart Lock	RFID Lock	DCM	Proximity Attack	Tag Emulation Tag Cloning
WiFi Router	Fast FW150RM	DCM	Proximity Attack	De-authentication

4.1 Physical Attacks: Attacks on IP Webcams and OFo bikes

In this section, we describe the analysis of some commercial webcams against physical attacks. In particular, we describe how default login settings of IoT devices and the mobile phone number may be used by the attackers to access the device illicitly, or to reset the password. Note that these attacks also work on similar classes of IoT devices, such as OFo bikes app [5,7]. Here, we report the results of the analysis of the Qihoo 360 camera[4] (360Camera), which is

[4] `http://jia.360.cn/`

designed by Qihoo 360 Technology Co. Ltd. in China. This camera is Qihoo's premier security camera, and includes several home security options, and is controlled by a mobile app. The camera offers real-time live streaming, including video recording. Text alerts can be set up when the camera detects movements, whereas the "Homewatch" function can send alerts to user's phone when the camera detects movements in a specific "anti-theft area" so the user is alerted, for instance, when doors and windows might be unprotected. Regarding its security, the camera app uses the mobile phone number as the username: hence, if the owner forgets the password, she can reset the password by requesting it on the website, and the mobile will receive a PIN so that the user is granted the right to reset the password. Note that the 360Camera app can support sharing the real-time and history videos with family people and other friends, which is a privacy-sensitive feature.

Initial Settings. The 360Camera mobile app is essential for camera setup and operation. This app is available for Android from Google Play, or for iOS from the Apple Store, which makes the whole process very easy and trusted for users. After the installation of this app on user's smartphone, the user uses the phone number as the username and sets a password for connecting to the web camera. The user sets the login process as default settings, so that it does not need to input the password and username again in the future.

Default Login Attack. In an example scenario, Alice loses her mobile phone or this has been left unattended for some time. The attacker's goal is to access the phone for a short amount of time to be able to modify the camera app setting to spy on Alice's privacy through the cam. In this scenario, after the attacker gets the mobile phone, he/she can login to the 360Camera app and share the camera's feeds with anyone he/she chooses. Note that this attack works only if the phone is unlocked, otherwise, as described in the "Reset Password Attack" (in the following), the attacker can remove the SIM card, insert it briefly into another phone to perform the reset password attack. Figure 2 shows the default login attack of 360Camera. The feed function is given by the Camera, as shown in Figure 2 titled "Invite Family to View". If the feeds have been shared with others, there is a list shown as "1 invited", where the number "181*****586" is the phone number of the attacker. In addition, the camera can be set as a public camera, so that everyone in the list can see the video.

Reset Password Attack. If the mobile phone is lost or the attacker gets it (as described in the previous attack), the attacker can remove the SIM card and insert it into his or her phone[5]. Since the number of the SIM card can be easily obtained by dialling another number, this allows the attacker to get the web camera's username, and he/she can obtain the PIN for resetting the password through the online reset password function. Then, the attacker obtains the access rights to access the camera's feed, and the attacker might even share it with

[5] We assume the SIM card is not locked, as over %60 people do not use the SIM lock functionality to restrict removing the SIM to another phone [15].

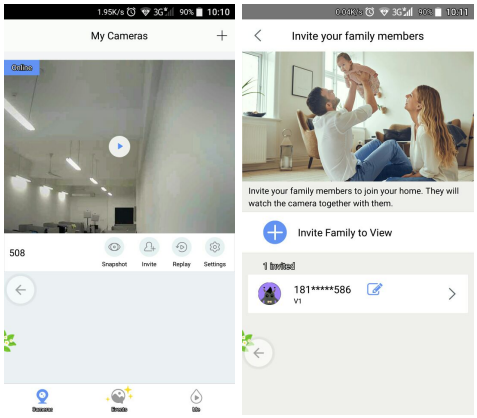

Fig. 2. Owner's APP: Default Login and Inviting Others to Access the Camera

others. In the tested environment, we have found that even if the legitimate owner has changed the login password, the shared users can still login to and get the real-time video of camera. In addition, the owner's app does not receive any notification that there is an app running in a third-party smartphone (that has been granted the sharing rights) that is accessing the camera. All the attacks on 360Camera have been reported to manufacturer, which believes there is little security risk while the phone number is used as a username to login the app. However, the manufacturer has produced an updated version that clearly displays the sharing function to the owner through a dynamic tab at the top left corner of app to show who has been granted access to the video feeds. The manufacturer is also considering the possibility of asking the owner to re-grant the same privileges to users when the password is reset.

We also have performed similar attacks on OFo bikes (China's "Uber for bikes")[6], similar to the one discussed previously on the 360Camera. In particular, if the mobile phone is lost or the attacker gets it, and if the phone has no default login setting, the attacker can remove the SIM card and insert it into his or her phone. When the attacker logs in to the OFo app, he/she can login successful if the phone is configured with the default login and gets the unlock code for entering the bike number. In addition, the attacker can re-obtain the verification code by using the phone number. To this end, OFo includes a QR code on the plate to allow users to retrieve the bike's number by scanning the code with a smartphone. However, an attacker can create a fake QR Code for this bike, and stick on top of the real QR code. Then, the user that will scan this QR Code with their mobiles phones will be redirected to a fake website that may request downloading a trojanized app that is similar to OFo app. Figure 3 shows this attack: here, we can see that an attacker can stick a transparent QR code to

[6] `http://www.ofo.so/`

replace the bike's QR code, and when the user scans the code she will download a fake app update, which gives an attacker remote access to it [21].

Fig. 3. Malicious QR Code Attack

4.2 Proximity Attacks: Attacks on RFID-based Smart Lock

RFID devices are widely used in smart-lock systems for security protection, and also because they are very convenient for users. When someone loses the RFID key, he/she just needs to elect a new RFID tags as the new key. RFID can also support access control very easily. However, RFID tags used in these systems are often vulnerable to various attacks, such as clone and relay attacks. Many researches have showed generic attacks for RFID systems, but little practical attacks for actual smart-lock system have been performed so far. In this section, our focus will be on attacks for gaining access to a restricted area by either cloning or emulating an access RFID tag. We will describe the following practical attacks on RFID smart-lock system: tag emulating, and tag cloning.

Tag Emulation. Because many RFID systems lack some security considerations, the default keys for each sector of tag often do not change, which can be used to get all of the data of the tag. Here, we describe how a default key attack is performed, and how all the tag data can be obtained. In this scenario, the tag can be emulated by a device that presents itself as a tag to the reader. A tag emulating device is implemented and can be bought online [8]. In our experiments, we have used the ACR 122U reader together with LIBNFC tools to emulate a tag. LIBNFC [23] is an open-source C library implementation for Near Field Communication (NFC) devices providing NFC Software Development Kit and Programmable API that can be used by RFID and NFC applications. Some of the important features of this library are: support for ISO 14443-A/B modulation, MIFARE Classic and Sony Felica protocol implementation, and ability to transform an USB-based NFC hardware device into a reader or tag. When the tag is emulated, we can use the ACR 122U reader to unlock the smart-lock.

Tag Cloning. The tag data can be maliciously read by some devices, such as PN532 [10] or ACR 122 [14] and Proxmark III [12], and dumped into a file. Then, the content of this dump file can be revised according to specific access requirements and could be written to a writeable UID tag. In detail, Proxmak III is used to sniff the communication information between the tag and the reader, and can produce the tag content to a dump file. ACR 122U reader can use specific software to write the dump file to a writeable tag. Then the cloned tag has the same information as the legal tag. An attacker can then use this cloned tag to unlock the smart-lock.

4.3 Proximity Attack: Attack on Smart-Home WiFi router

Currently, several WiFi devices are used in smart-home environments and they need to connect to a WiFi router to access the Internet. In this section, we discuss how de-authentication (deauth) attack [6] for WiFi router can be used to disconnect smart-home IoT devices from the Internet. The 802.11 WiFi protocol contains deauthentication frame which is used to disconnect clients from a WiFi network. Attackers can send a deauthentication packet to the WiFi transmitter station at any time using the spoofed source address of the wireless AP. The attacker does not need to know the password to get into the WiFi network, as he/she needs just to be in the WiFi signal range. For this attack, we have used a Pocket 8266 NodeMCU, which is an open source hardware that can be revised as an attack device. The attack can be extended to external networks by using an Unmanned Aerial Vehicle (UAV) [22] to perform the attack when the attacker is out of the WiFi local network. Fig. 4 depicts the steps of this attack. In detail, a mobile phone (Phone 2) is connected to NodeMCU, which we call a *proxy-attack-device*. Then, the UAV first brings this device into the WiFi area and then it leaves the WiFi area, so that the proxy-attack-device is left in the WiFi area. Note that the mobile phone (Phone 2) gives power supply to Node MCU, and the Node MCU is controlled by another (remote) mobile phone (Phone 1) via Phone 2. The attack is now performed by the proxy-attack-device controlled by Phone 1.

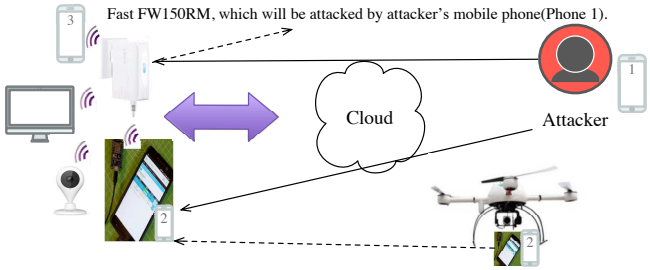

Fig. 4. WiFi Attack

The steps to perform the attack are described in the following:

Initial Settings and Attack. In the experiment, the WiFi device (Fast FW150RM[7]) is configured to use the default settings of the manufacturer. Note that all the smart-home IoT devices are connected to this router. Then, the next steps are: (i) UAV brings the attack device, which includes a mobile phone (Phone 2) connected to Pocket 8266 NodeMCU, to the target WiFi network. The attacker uses another mobile phone (Phone 1) browser to connect to a web server running in NodeMCU. Therefore, the attacker can use it to control Pocket 8266 NodeMCU to attack the WiFi devices (see Fig. 5); (ii) the attacker scans the available WiFi hotspots, as shown in Fig. 5; he/she selects one WiFi hotspot, and then it scans which IoT devices are connected to the WiFi hotpot (Fig. 5 shows that five clients were found in this range during our experiments); finally, (iii) the attacker selects one client to start the attack, which is shown in Fig. 5. Then, the selected device is disconnected from the WiFi after it has been attacked.

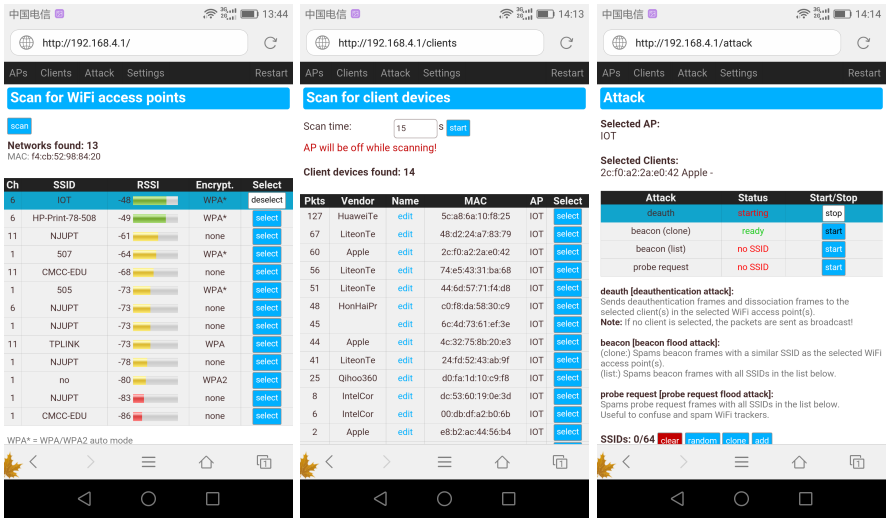

Fig. 5. Attack on Smart-Home WiFi Router

5 Conclusion

In this paper we have examined the security of commodity IoT devices by performing physical and proximity attacks. The attack scenarios and experimental results show that IoT devices attacks are practical and may seriously impact the safety and privacy of users. In our analyses, we have seen that many commodity

[7] http://www.fastcom.com.cn/

IoT devices do not address basic vulnerabilities under different attackers capabilities. In detail, we have found that IP cameras may expose private feeds to third parties, smart-locks can be easily opened, and smart-home WiFi routers may be exploited in WiFi range, and in extended range with the help of UAVs. Many of the attacks presented in this paper highlight the need to improve the usability of IoT applications in order to improve their security. These attacks also clearly demonstrate that manufactures should design IoT security carefully by consider different threat models, such as physical and proximity attacks.

Acknowledgments

This work is financially supported by Jiangsu Government Scholarship for Overseas Studies, the National Natural Science Foundation of P. R. China (No.61373017, No.61572260, No.61572261, No.61672296, No.61602261), the Natural Science Foundation of Jiangsu Province (No.BK20140886, No.BK20140888), Scientific and Technological Support Project of Jiangsu Province (No. BE2015702, BE2016185, No. BE2016777), China Postdoctoral Science Foundation (No. 2014M551636, No.2014M561696), Jiangsu Planned Projects for Postdoctoral Research Funds (No.1302090B, No.1401005B), Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX17_0798).

References

1. The internet of things has started (April 2016), http://www.mycustomer.com/community/blogs/corelynx/the-internet-of-things-has-started-have-you-joined-the-iot-bandwagon
2. There will be 24 billion iot devices installed on earth by 2020 (June 2016), http://uk.businessinsider.com/there-will-be-34-billion-iot-devices-installed-on-earth-by-2020-2016-5?r=US&IR=T
3. Brickerbot: "the doctor's" pdos attack has killed over 2 million insecure devices (April 2017), https://fossbytes.com/brickerbot-malware-pdos-attack-iot-device/
4. Brickerbot, the permanent denial-of-service botnet, is back with a vengeance (April 2017), https://arstechnica.com/security/2017/04/brickerbot-the-permanent-denial-of-service-botnet-is-back-with-a-vengeance/
5. Chinese bike-sharing start-up ofo says it's now worth more than $2 billion (April 2017), http://www.cnbc.com/2017/04/17/ofo-chinese-bike-sharing-start-up-says-its-now-worth-more-than-2-billion.html
6. esp8266_deauther (July 2017), https://github.com/spacehuhn/esp8266_deauther#supported-devices
7. Look out cambridge, here comes ofo – china's 'uber for bikes' (April 2017), http://www.wired.co.uk/article/chinese-bike-sharing-company-ofo-is-coming-to-cambridge-in-the-uk
8. Rfid emulator (July 2017), http://www.instructables.com/id/RFID-Emulator-How-to-Clone-RFID-Card-Tag-/
9. Bertino, E., Islam, N.: Botnets and internet of things security. Computer 50(2), 76–79 (2017)

10. Coskun, V., Ozdenizci, B., Ok, K.: A survey on near field communication (nfc) technology. Wireless personal communications 71(3), 2259–2294 (2013)
11. Fernandes, E., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 636–654 (May 2016)
12. Garcia, F.D., de Koning Gans, G., Verdult, R.: Tutorial: Proxmark, the swiss army knife for rfid security research. Technical Report, Radboud University Nijmegen (2012)
13. Ho, G., Leung, D., Mishra, P., Hosseini, A., Song, D., Wagner, D.: Smart locks: Lessons for securing commodity internet of things devices. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. pp. 461–472. ASIA CCS '16, ACM, New York, NY, USA (March 2016), http://doi.acm.org/10.1145/2897845.2897886
14. Huang, C.H., Chang, S.L.: Study on the feasibility of nfc p2p communication for nursing care daily work. Journal of Computers 24(2), 33–45 (2013)
15. Imgraben, J., Engelbrecht, A., Choo, K.K.R.: Always connected, but are smart mobile users getting more security savvy? a survey of smart mobile device users. Behaviour & Information Technology 33(12), 1347–1360 (2014)
16. Jerkins, J.A.: Motivating a market or regulatory solution to iot insecurity with the mirai botnet code. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC). pp. 1–5. IEEE (January 2017)
17. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: Ddos in the iot: Mirai and other botnets. Computer 50(7), 80–84 (2017)
18. Min, B., Varadharajan, V.: Design and evaluation of feature distributed malware attacks against the internet of things (iot). In: 20th International Conference on Engineering of Complex Computer Systems (ICECCS). pp. 80–89. IEEE (December 2015)
19. Ronen, E., Shamir, A.: Extended functionality attacks on iot devices: The case of smart lights. In: IEEE European Symposium on Security and Privacy. pp. 3–12. IEEE (March 2016)
20. Sgandurra, D., Lupu, E.: Evolution of attacks, threat models, and solutions for virtualized systems. ACM Comput. Surv. 48(3), 46:1–46:38 (Feb 2016), http://doi.acm.org/10.1145/2856126
21. Sivaraman, V., Chan, D., Earl, D., Boreli, R.: Smart-phones attacking smart-homes. In: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 195–200. ACM (July 2016)
22. Valavanis, K.P., Vachtsevanos, G.J.: Handbook of unmanned aerial vehicles. Springer Publishing Company, Incorporated (2014)
23. Verdult, R., de Koning Gans, G., Garcia, F.D.: A toolbox for rfid protocol analysis. In: Fourth International EURASIP Workshop on RFID Technology (EURASIP RFID). pp. 27–34. IEEE (September 2012)