Vulnerabilities in IoT Devices for Smart Home Environment

Luís Costa[1] João Paulo Barros[1,2] and Miguel Tavares[1,2,3]
[1]*Instituto Politécnico de Beja, Beja, Portugal*
[2]*Center of Technology and Systems (CTS) - UNINOVA, Portugal*
[3]*Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento - Lisbon (INESC-ID), Portugal*

Keywords: Internet of Things, Smart Home, IoT, IoT Security, Information Security, IoT Vulnerabilities.

Abstract: Recently, consumers have seen multiple products being advertised as smart home. These products promise to make our homes more comfortable, safe, automated, and remotely controlled. To this new reality of processing information it was given the name IoT (Internet of Things). Many news headlines have been published exposing serious security vulnerabilities in many IoT devices, with some of them being exploited to make one of the largest DDoS attacks recorded. In this paper we present a method developed with the purpose of identifying high risk vulnerabilities in smart home IoT devices, giving application examples of actual vulnerabilities found in two commercially available devices. This method uses several open source tools to identify vulnerabilities in some of these IoT devices. Besides, we will also present some topics related to the main threats and vulnerabilities that affect smart home IoT devices.

1 INTRODUCTION

IoT devices are gradually peaking the interest of consumers. Several studies predict an almost exponential growth on the number of IoT devices, totalling between 20 and 30 billion by 2020 (calsoft, 2018), with the smart home IoT market also growing in revenue. According to consumers (Perez, 2018), security problems are the main factor that is slowing down IoT growth.

In the past years, IoT devices have been seen many times as insecure devices. In 2015 Symantec (Barcena and Wueest, 2015) released a report on which major vulnerabilities on many smart home IoT devices were identified. In 2016 the Mirai botnet was responsible for one of the biggest DDoS attacks ever recorded, using more than 600 thousand devices, most of them IP cameras (Antonakakis et al., 2017). Two online databases https://www.exploitee.rs/ and http://www.hardwaresecurity.org/iot/database/) describe vulnerabilities on more than 200 devices, where about half of them are smart home IoT. On DEF CON 2017, one of the largest security and hacking conferences, 47 new vulnerabilities were revealed on 23 devices from 21 different manufacturers.

This number of vulnerabilities may lead consumers to think that all IoT devices have serious security vulnerabilities, which, however, is not entirely true. Even if a device has vulnerabilities, that does not mean it is insecure. If a vulnerability can only be explored with physical access to the device, it represents a small risk for the average consumer. With these factors in mind, it is crucial to identify vulnerabilities that can be explored remotely, especially through the internet.

This paper presents a method to identify vulnerabilities in smart home IoT devices. This method represents a practical approach that uses open source tools to identify vulnerabilities that can be explored through TCP/IP networks. The method can even be applied to devices that aren't smart home, allowing manufacturers to identify the most explored vulnerabilities in cyberattacks, with a low cost method.

Additionally, the paper also presents a vision of what IoT and smart home IoT are, by analysing aspects of its ecosystem, main vulnerabilities and their real risk. Through these topics we will identify the higher risk threats to data protection and to private home security.

2 RELATED WORK

To the best of our knowledge, there is no method that allows the identification and exploitation of vulnera-

Costa L., Barros J. and Tavares M.
Vulnerabilities in IoT Devices for Smart Home Environment.
DOI: 10.5220/0007583306150622
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 5th International Conference on Information Systems Security and Privacy)*, pages 615-622
ISBN: 978-989-758-359-9

bilities in smart home IoT devices like the one here presented. There are a few similar ones, but they are based on automated scanning tools, that have trouble if they cannot correctly identify the device (Visoottiviseth et al., 2017)(Leite Da Silva, 2017). There are others that focus on hardware and non TCP/IP networks (like Bluetooth and Zigbee) (Gupta, 2017), and LLN networks that rely on IPv6 not currently adopted by smart home IoT systems (Martins, 2018).

3 SMART HOME IOT DEVICES

Smart home is one of the many components that composes IoT. According to one definition, a smart home "*is a house or living environment that contains the technology to allow home devices and systems to be controlled automatically*" (Bing et al., 2011). Smart home is a concept that has been around for quite some time, but IoT made it more attractive.

There is still no consensual definition for the smart home in the IoT era (Schiefer, 2015). For this reason it is important to present the definition adopted in this paper: *smart home* is "*a residence equipped with a communications network, linking sensors and domestic devices and appliances that can be remotely monitored, accessed or controlled, and which provides services that respond to the needs of its inhabitants*" (Ghaffarianhoseini et al., 2016).

3.1 From Architecture to Reality - The Security Perspective

At some point we need to step out from the conceptual models of architectures and look at the implementations adopted by the manufacturers. Because there are no standard practices applied (Al-Qaseemi et al., 2017), sometimes manufacturers'implementations do not follow the architectures. This scenario may change by global agreement of stakeholders in IoT development, with efforts like the Code of Practice for Consumer IoT Security (Department for Digital, Culture, Media and Sport (DCMS), 2018). Since we are presenting a practical method, we need to define how the current smart home IoT devices implement the theoretical architectures. To this implementation we will call smart home IoT ecosystem.

This ecosystem was built by studying the design implemented by several manufacturers and also according to the work of experts that conduct penetration tests on IoT devices (Pesce, 2017)(Gupta, 2017). This ecosystem is divided into five components (see Fig. 1):

- **Hardware -** Includes all the physical components of the ecosystem including the firmware;
- **Web Application -** It can be used to configure, control or access a device. In many cases, it is hosted in a web server that resides in the device itself;
- **Mobile Application -** These applications are gradually replacing web applications since they present more features and are more practical for the user;
- **Network and Cloud -** It includes all communication between devices, the web and mobile application, and the cloud;
- **API -** The Application Programming Interface plays a crucial role in IoT. It makes the integration and interoperability between devices, applications, and the cloud possible.

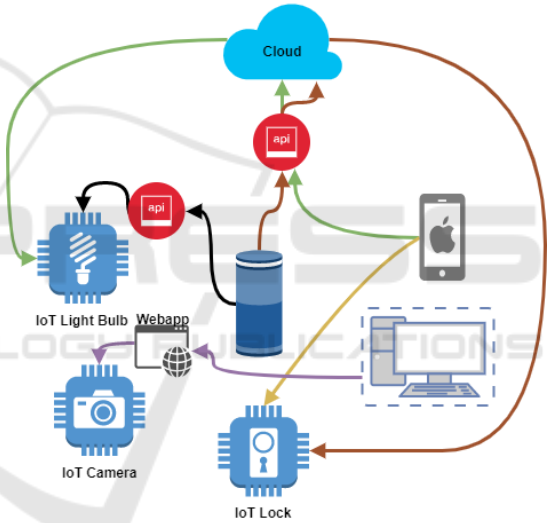

Figure 1: Interaction between components of the smart home IoT ecosystem.

4 SMART HOME IOT SECURITY

In this section we discuss security challenges, threats, and current vulnerabilities of smart home IoT devices.

4.1 Security Challenges

IoT systems, especially smart home, have many things in common with traditional computer networks, including some security problems. Within a home network we may have devices like computers, smartphones, and tablets that have lots of processing power and are constantly monitored by users. On the other hand, IoT devices have hardware and design

limitations that prevent traditional security measures from being easily implement. The following are some of these limitations(Haritha and Lavanya, 2017):

- IoT devices often use low speed CPUs and are battery powered. Contemporary cryptography algorithms require fast computation;
- IoT devices usually have memory restriction and use low bandwidth communications. Current security schemes were not designed to support these restrictions;
- In some cases, the installation of security patches may be impossible. IoT devices use lightweight operating system that might lack modules to receive and integrate new codes or libraries;
- The IoT universe is made of different types of devices ranging from computers to RFID cards using different wireless protocols. It is hard to find a security solution that is capable of accommodating all these devices.

4.2 Top Vulnerabilities

One of the differences between traditional computer networks and IoT ones are the threats they face. Many times, smart home IoT attacks have consequences that transcend the boundaries of cyberspace and affect people lives and the safety of their homes (Denning et al., 2013)(Soltan et al., 2018). But many of these attacks explore vulnerabilities that are well known by IT Administrators and security experts. These vulnerabilities gained an all new ”life” within IoT.

Within this big media coverage that preach an IoT apocalypse (Newman, 2018)(Hiner, 2018) it is important to identify the vulnerabilities that pose the biggest risk to the safety of smart home IoT and their real impact. Some may affect a big number of devices but are not very easily exploited (Dorsey, 2018), therefore have a low risk. One of the biggest treats to smart home networks are automated attacks. They have not a specific home as a target, but a type of device or a specific network service. With these aspects in mind, the vulnerabilities that are more likely to be exploited on smart home IoT attacks are the following(OWASP, b)(Khan and Salah, 2018)(Embedi, 2018):

- Lack of security updates;
- Insecure web application and services authentication;
- Insecure services exposed to the internet;
- Insecure network communications.

4.3 Penetration Testing

IoT devices are bringing new security challenges to home network security (see section 4.1). A vulnerability in a smart door lock can lead to a robbery or to other violent crimes (Denning et al., 2013). So the need to test home network security can increase with the growing adoption of smart home IoT devices. Penetration testing is a well know technique applied by ethical hackers, that makes use of some of the tools employed by malicious hackers, in order to simulate attacks and identify security vulnerabilities.

The method to identify vulnerabilities in smart home IoT devices that we propose is based on the PTES Standard (http://www.pentest-standard.org/). This standard needed to be adapted since it lacks some specific steps related to IoT smart home systems. For example, some of the information gathering techniques suggested by the standard are targeted to collect information about persons, like a company CIO. In a smart home IoT ecosystem (see section 3.1) the human factor is not that important so we need to gather information from other sources like the devices firmware or mobile application source code, to understand how devices interact.

5 IDENTIFYING VULNERABILITIES

This section presents the proposed method. The respective results are presented in Section 6.

5.1 Introduction

The previous sections provided useful information in order to understand the ideas behind this method. Since we are still in the beginning of the IoT era and manufacturers are still waiting for standardisation of architectures and technologies (Briodagh, 2018), there is no universal recipe to identify vulnerabilities in IoT systems. By having a general understanding of the ideas, challenges, current architectures, threats, and vulnerabilities, this method becomes more easily adapted to different and future smart home IoT ecosystems.

5.2 Method Structure

The method is divided into different stages and procedures represented by an activity diagram (Omg, 2017) (see Fig. 2).

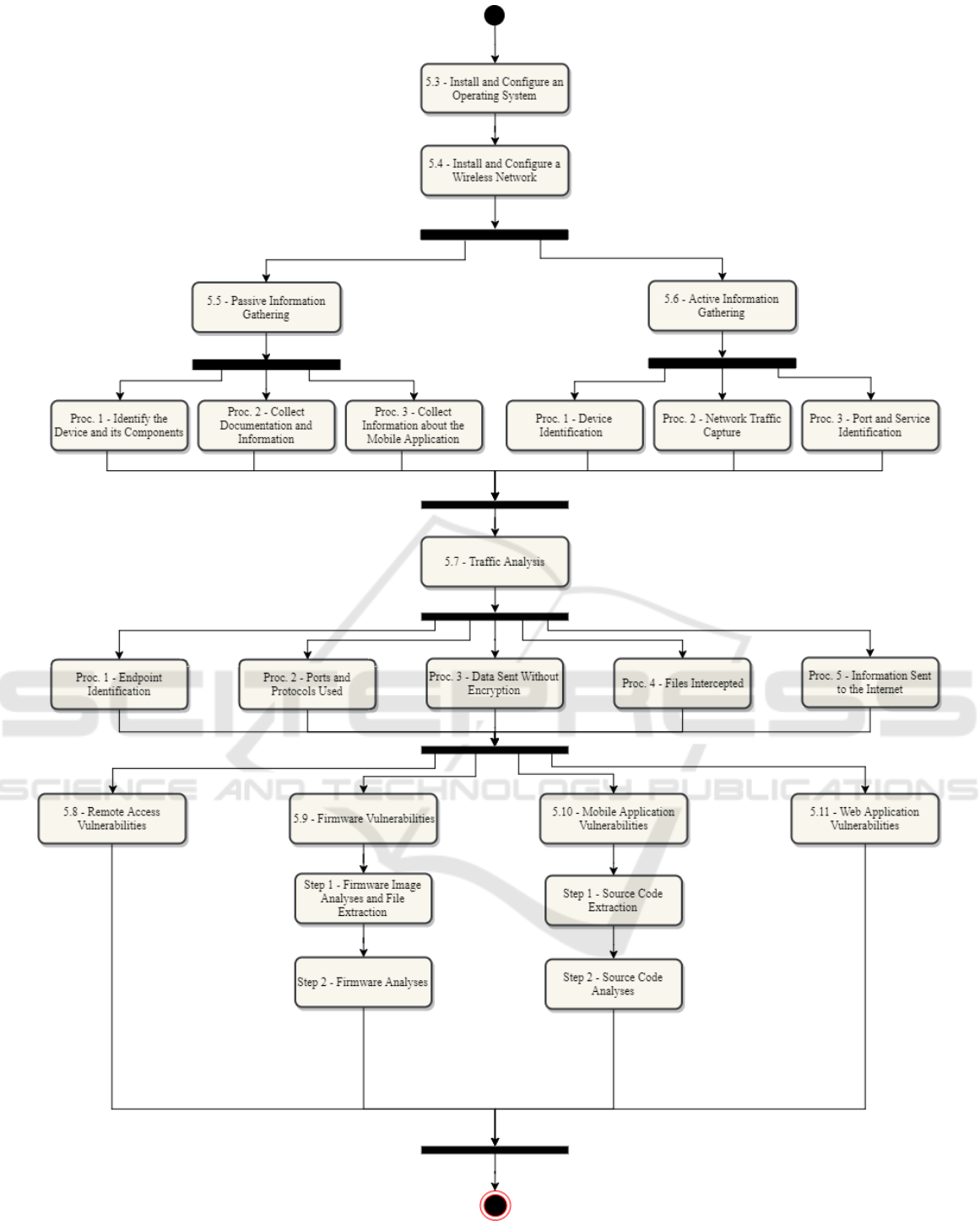

Figure 2: Activity diagram of the proposed method.

All the procedures and stages at the same level can be executed in parallel, but only after all are completed we can move to the next level. The ones that cannot be executed for some reason (e.g. the device does not have a web interface) should be ignored together with the respective arcs; this would imply additional paths in the diagram, yet, for better readability, they are not presented. To make interpreting the method easier, the numbering in the diagram matches the ones of the following subsections.

5.3 Install and Configure an Operating System (OS)

Correctly configuring a test environment is a key aspect in order to ensure that the tools we use work properly. The easiest way is to use a pre-build operating system (OS) dedicated to ethical hacking. These include some of the tools that will be used later in the method. There are many OS's available, but it is import to choose one that is frequently updated. The ones we tested and therefore recommend are the following:

- Kali Linux (https://www.kali.org)
- Parrot OS (https://www.parrotsec.org).

We also recommend to install the operating system in a virtual machine. It is more practical and there is there is no need for a dedicated computer just for this purpose. We will call this machine "HackOS".

5.4 Install and Configure a Wireless Network

Most smart home IoT devices require a wireless network to communicate. By using a specific network for the method we are reducing the number of communications to analyse, and not taking the risk of exposing other network devices to security risks. To make intercepting communications easier we recommend creating a WiFi Access Point in the "HackOS" machine.

5.5 Passive Information Gathering

This stage focuses on collecting information about the smart home IoT device and its ecosystem. In this stage no communication should be made with the IoT device or its ecosystem. The point here is to determine what kind of information is publicly available. This stage is divided into three procedures:

5.5.1 Proc. 1 – Identify the Device and its Components

Before starting to collect more information about the device it's important to identify the brand, model and manufacturer. If possible, the device should be dismantled in order to identify any part numbers, logos or other inscriptions.

5.5.2 Proc. 2 – Collect Documentation and Information

The more information we collect on a device, the bigger the chances are of understanding the device and how it interacts with its ecosystem. Look for user manuals, API documentation, open source projects, and firmware installationfiles;

5.5.3 Proc. 3 – Collect Information about the Mobile Application

Most IoT device have a mobile application used to control and collect data from them. On the Android operating system it is possible to obtain an installation file directly from an online repository. The following are the most popular ones:

- APKmirror (https://www.apkmirror.com/);
- Aptoide (https://pt.aptoide.com/);
- APKpure (https://apkpure.com).

5.6 Active Information Gathering

Active information gathering means collecting information about the smart home IoT device and its ecosystem by communicating and intercepting their network traffic. This stage is divided into three procedures presented in the next subsections.

5.6.1 Proc. 1 – Device Identification

This procedure is about identifying the devices' IP and MAC addresses in the network. If the device has a WiFi network for configuration we should try to connect to it before making any configurations. Some devices expose different ports or services or have a different behaviour before they are configured.

To identify the IP and MAC address of the device we recommend using the tool **Netdiscover** (https://github.com/alexxy/netdiscover).

5.6.2 Proc. 2 – Network Traffic Capture

By capturing the communications sent and received by the device and its ecosystem we can extract valuable information later on. Since some IoT devices communicate in a different way based on their state, it is important to capture the devices network traffic in different states (Pesce, 2017):

- Booting up, without configuration and in standby;
- Communicating with the mobile and web applications;
- During a firmware update;
- Without internet connection.

To capture the network traffic we recommend using the tool **Wireshark** (https://www.wireshark.org/).

5.6.3 Proc. 3 – Port and Service Identification

The final procedure of this stage allows us to enumerate the network ports that are open on the device and what services they expose. Some devices have dozens of open ports (Loi et al., 2017) that expose services like SSH or Telnet.

The recommended tools for this procedure are **Nmap** (https://nmap.org/) and **Zenmap** (https://nmap.org/zenmap/).

5.7 Traffic Analysis

This section is about analysing the network captures made in the previous section. The goal of this analysis is to determine how the different ecosystem components interact with each other, and what kind of information is sent and received. The following are the key aspects to identify at this stage:

- Proc. 1 – Endpoint Identification: Identify IP address, domain and location of all endpoints;
- Proc. 2 – Ports and Protocols: What network ports and protocols were used;
- Proc. 3 – Data Sent Without Encryption;
- Proc. 4 – Files Intercepted;
- Proc. 5 – Information Sent to the Internet;

In order to analyse the network captures we recommend using the tool **Wireshark**.

5.8 Remote Access Vulnerabilities

Some IoT devices have a remote access protocol available, like Telnet or SSH. In fact many devices that formed the Mirai bootnet (Antonakakis et al., 2017) were infected because they had their Telnet service exposed to the internet with default credentials. The most common attacks to these kind of services are Brute-Force or Dictionary-Attacks.

To test if the remote access service is vulnerable to any of the attacks mentioned or others, we recommend using the tools **Medusa** (https://github.com/jmk-foofus/medusa), **Ncrack** (https://github.com/nmap/ncrack) or **Hydra** (https://github.com/vanhauser-thc/thc-hydra).

5.9 Firmware Vulnerabilities

Exploring an IoT device firmware is one the most common ways to identify its vulnerabilities. Since some of these devices feature a striped down version of a Linux OS, their file structure and services are very similar to the desktop version. One of the most common vulnerabilities found is the exposure of sensitive information (OWASP, b) like login credentials. We propose a two step approach in order to identify sensitive information in the firmware.

5.9.1 Step 1 – Firmware Image Analysis and File Extraction

The objective of this procedure is to extract the firmware file system from the firmware image, typically a BIN file. Some vendors have these files available for download from their websites, or we can try to get it by capturing the network traffic during a firmware update (some devices use plain HTTP on update). There are other methods that involve extracting the firmware directly from the device storage, but they require physical access to the device.

To extract the firmware files from the image we recommend using **binwalk** (https://github.com/ReFirmLabs/binwalk).

5.9.2 Step 2 – Firmware Analysis

After getting access to the files in the firmware we can start to look for information that can help us identify some vulnerabilities and understand the device behaviour. The information to look for is the following (Gupta, 2017)(OWASP, a):

- **Login Credentials -** Usernames and passwords left on the code;
- **Backdoors -** Most commonly Telnet or SSH services;
- **URL's -** Firmware or source code repositories, unauthenticated cloud or API connectios, and other;
- **Criptographic Keys -** Symmetric keys in the source code or in a file;
- **Cypher Algorithms -** Information about encryption algorithms can help to decrypt communications;
- **Authentication Mechanisms -** Details about the API, web application, or other authentication processes.

To ease the process of going through all the firmware files we recommend using a tool called **Firmwalker** (https://github.com/craigz28/firmwalker). This tool searches for specific words in files (like "password" or "admin"), extensions, file types and others. After running the tools it produces a report of all the files that matched the search criteria and even other information like IP and email addresses.

5.10 Mobile Application Vulnerabilities

Some mobile applications also expose sensitive information that may help explore other vulnerabilities on the ecosystem. Besides this, they can also have vulnerabilities of their own. This stage will only focus Android applications.

In order to perform a full vulnerability analysis in mobile applications there is a large amount of specific knowledge about the operating system and mobile application development that is needed. But even without that knowledge we can use tools to identify vulnerabilities. This stage is divided into two steps and offers a simple way to identify vulnerabilities.

5.10.1 Step 1 – Source Code Extraction

Many times, Android applications source code can be extracted from the APK installation file. This file can be obtained from one of the repositories mentioned in Proc. 3 of stage 5.5. The files are then reverse engineered into Java classes that can be analysed.

In order to extract the source code we recommend using **Jadx** (https://github.com/skylot/jadx) or **QARK** (https://github.com/linkedin/qark). The former can only be used to extract the source code, but the latter includes other features covered in the next step.

5.10.2 Step 2 – Source Code Analysis

The process of analysing the source code of mobile applications has two purposes: (1) to better understand the interaction of the different ecosystems components and (2) to find vulnerabilities within the mobile application.

This process is very similar to the one used in step 2 of the previous stage. In fact we recommend running the tool **Firmwalker** in the source code of the mobile application, since it may have the same kind of information leakage vulnerability. Another tool we recommend using is **QARK** since it runs a static code analysis algorithms in order to find vulnerabilities in the mobile application.

5.11 Web Application Vulnerabilities

Web applications are still present in some IoT devices. Their vulnerabilities can be identified with the same tools used for common websites or non IoT web applications. IoT devices web applications exposed to the internet are a serious threat to IoT security if they are not properly secured.

Since the biggest threat to smart home IoT security are automated attacks (see section 4.2), we can use one of these automated scanning tools to identify some of the web application vulnerabilities like the popular **OWASP ZAP** (https://github.com/zaproxy/zaproxy).

6 METHOD IMPLEMENTATION RESULTS

In order to test the method we applied it to two devices, a smart light bulb, the Yeelight Led Color (Device 1) and, the DVR Dahua DHI-XVR5108H-4KL-8P (Device 2). To the first device we applied most of the method stages excluding the ones related to the device firmware analysis and web application (the device does not have one). For the second one we only applied the passive information gathering and firmware vulnerability stages.

Device 1 has a bare-metal OS and the firmware source files could not be extracted. Nevertheless we were able to obtain the firmware image through the active information gathering stage.

We also performed an attack to the device using a python library (https://github.com/rytilahti/python-miio identified in the passive information gathering stage. This library allowed us to control the device trough its WiFi configuration network, without any authentication. This attack explores two of the vulnerabilities identified. The first one is the lack of authentication on the configuration process, since no password is requested for the WiFi configuration network. The second one is an unauthorised control vulnerability, since the device exposed an authentication token that allows its control. The attack is limited to a 30 minute time window, since after that period the network is turned off.

We were also able to find credentials in the mobile application source code, that appear to be for a mesh network authentication as well as some potential vulnerabilities in the mobile application reported by the QARK tool.

On Device 2, we only had access to its firmware (downloaded from the manufacturer website). This device was used to test if the method could be applied without having physical access to the device. We found critical information leakage vulnerabilities in the device firmware. The device had default administrator login credentials in plain text on a configuration file, possibly for the device web application. We ran a search in the Shodan database and found more then 14000 IP addresses that expose this device web application to the internet.

7 CONCLUSION AND FUTURE WORK

We presented a practical method supported by open source tools that can identify high risk vulnerabilities present in smart home IoT devices. By following this method manufacturers and advanced users can test if their devices are vulnerable to the most common vulnerabilities that are being exploited in cyberattacks. Applying this method we were able to identify vulnerabilities in two test devices. By exploring some of those vulnerabilities, we were able to control one of the devices without authentication. We were also able to identify vulnerabilities without physical access to a device. These vulnerabilities can potentially be exploited through the internet since thousands of devices were identified in the Shodan database.

For future work we would like to develop specific tools that could help automate the traffic analysis section of the method. It would also be interesting to apply the method to different IoT ecosystems to further test its applicability or even to introduce new stages or steps in the proposed method.

REFERENCES

Al-Qaseemi, S. A., Almulhim, H. A., Almulhim, M. F., and Chaudhry, S. R. (2017). IoT architecture challenges and issues: Lack of standardization. In *FTC 2016 - Proc. of Future Technologies Conference*, San Francisco, CA, USA. IEEE.

Antonakakis, M., April, T., Bailey, M., Bursztein, E., Cochran, J., Durumeric, Z., Alex Halderman, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y., Antonakakis Tim April, M., Bernhard Elie Bursztein, M., Cochran Zakir Durumeric Alex Halderman Luca Invernizzi, J. J., Kallitsis, M., Kumar, D., Lever Zane Ma, C., Mason, J., and Sullivan Kurt Thomas, N. (2017). Understanding the Mirai Botnet. In *SEC'17 Proc. of the 26th USENIX Conference on Security Symposium*, Vancouver, BC, Canada. USENIX Association Berkeley.

Barcena, M. and Wueest, C. (2015). Insecurity in the Internet of Things. Technical report, Symantec.

Bing, K., Fu, L., Zhuo, Y., and Yanlei, L. (2011). Design of an IoT-based smart home system. In *2011 2nd Int. Conf. on Intelligent Control and Information Processing*, Harbin, China. IEEE.

Briodagh, K. (2018). HomeGrid Forum Calls for Standardization and Certification of Devices.

calsoft (2018). Internet of Things (IoT) 2018-Market Statistics, Use Cases and Trends. Technical report, calsoft.

Denning, T., Kohno, T., and Levy, H. M. (2013). Computer security and the modern home. *Communications of the ACM*, 56(1).

Department for Digital, Culture, Media and Sport (DCMS) (2018). Code of Practice for consumer IoT security. Technical report.

Dorsey, B. (2018). Attacking Private Networks from the Internet with DNS Rebinding.

Embedi (2018). Reflecting upon OWASP TOP-10 IoT Vulnerabilities – Embedi.

Ghaffarianhoseini, A., Ghaffarianhoseini, A., Tookey, J., Omrany, H., Fleury, A., Naismith, N., and Ghaffarianhoseini, M. (2016). The Essence of Smart Homes: Application of Intelligent Technologies towards Smarter Urban Future. In *Creative Technologies for Multidisciplinary Applications*, chapter 14. IGI Global, Hershey, PA.

Gupta, A. (2017). *IoT Hackers Handbook*. Attify, 1 edition.

Haritha, A. and Lavanya, A. (2017). Internet of Things: Security Issues. *International Journal of Engineering Science Invention ISSN (Online)*, 6(11).

Hiner, J. (2018). New research: Most IoT devices can be hacked into botnets - TechRepublic.

Khan, M. A. and Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82.

Leite Da Silva, C. (2017). *Ragnar: Ferramenta para Pentest em dispositivos da Internet das Coisas*. PhD thesis, Universidade de Brasília.

Loi, F., Sivanathan, A., Gharakheili, H. H., Radford, A., and Sivaraman, V. (2017). Systematically Evaluating Security and Privacy for Consumer IoT Devices. In *Proc. of the 2017 Workshop on IoT Security and Privacy - IoTS&P '17*, Dallas, Texas, USA. ACM New York.

Martins, R. (2018). *Desenvolvimento de uma Framework para Investigação de Vulnerabilidades em Dispositivos de Internet das Coisas*. PhD thesis, Universidade Estadual de Londrina.

Newman, L. (2018). Millions of Google, Roku, and Sonos Devices Are Vulnerable to a Web Attack — WIRED.

Omg (2017). OMG Unified Modeling Language (OMG UML). Technical report.

OWASP. IoT Firmware Analysis - OWASP.

OWASP. OWASP Internet of Things Project - OWASP.

Perez, Y. (2018). Smart home device ownership 'to rise by 2022'.

Pesce, L. (2017). Sans webcast: I don t give one iota - introducing the iot attack methodology - youtube.

Schiefer, M. (2015). Smart Home Definition and Security Threats. In *Proc.- 9th Int. Conf. on IT Security Incident Management and IT Forensics, IMF 2015*, Magdeburg, Germany. IEEE.

Soltan, S., Mittal, P., and Poor, H. V. (2018). BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid. In *USENIX Security Symposium*. USENIX.

Visoottiviseth, V., Akarasiriwong, P., Chaiyasart, S., and Chotivatunyu, S. (2017). PENTOS: Penetration Testing Tool for Internet of Thing Devices. In *IEEE Region 10 Conference (TENCON)*.