ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Ingeniería de Telecomunicaciones y Máster en ciberseguridad

Análisis de Vulnerabilidades en Dispositivos HVAC

Autor
José Antonio Font García

Dirigido por
Gregorio López López
y
Rafael Palacios Hielscher

Madrid
Jul 2023

José Antonio Font García, declara bajo su responsabilidad, que el Proyecto con título **Análisis de Vulnerabilidades en Dispositivos HVAC** presentado en la ETS de Ingeniería (ICAI) de la Universidad Pontificia Comillas en el curso académico 2022/23 es de su autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: José Antonio Font García Fecha: 12 / 07 / 2023

Autorizan la entrega:

Directores del Proyecto

Gregorio López López

Fdo.: .... Fecha: 12 / 07 / 2023

Rafael Palacios Hielscher

Fdo.: .... Fecha: ...... / ...... / ....

V. B. del Coordinador de Proyectos

Nombre del Coordinador

Fdo.: .... Fecha: ...... / ...... / ....

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. José Antonio Font García **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra: Análisis de vulnerabilidades en dispositivos HVAC, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

(a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.

(c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.

(d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.

(e) Asignar por defecto a estos trabajos una licencia Creative Commons.

(f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

(a) Que la Universidad identifique claramente su nombre como autor de la misma

(b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

(c) Solicitar la retirada de la obra del repositorio por causa justificada.

(d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

(a) El autor se compromete a:

(b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

(c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

(d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

(e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a ..12.. de ...Julio........... de 2023......

ACEPTA

Fdo.: José Antonio Font García

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Ingeniería de Telecomunicaciones y Máster en ciberseguridad

Análisis de vulnerabilidades en dispositivos HVAC

Autor
José Antonio Font García

Dirigido por
Gregorio López López
y
Rafael Palacios Hielscher

Madrid
Jul 2023

Resumen

La necesidad de realizar análisis de amenazas en dispositivos HVAC (Calefacción, Ventilación y Aire Acondicionado) surge de los riesgos que aparecen por la introducción de capacidades inteligentes en estos. Uno de estos riesgos, el que constituye el foco de este trabajo, es la posibilidad de perturbar la red eléctrica a través de un ataque coordinado a dispositivos inteligentes de alto consumo con medidas de seguridad deficientes. El entorno de los dispositivos HVAC es relevante en este escenario tras haber experimentado un rápido desarrollo en la dirección de los dispositivos inteligentes como así lo evidencia la considerable oferta que se puede encontrar en mercados digitales como *Amazon*.

La necesidad de evaluar el riesgo en el ecosistema eléctrico europeo ha llevado al desarrollo de este proyecto, cuyo propósito principal es contribuir a este esfuerzo a través de la evaluación de la vulnerabilidad ante este tipo de ataque que corren los productos disponibles en el mercado europeo. Para conseguir esto de manera sistemática, se ha desarrollado una metodología de modelado de amenazas que posteriormente se ha aplicado para guíar las pruebas sobre un dispositivo pertinente.

Dicha metodología, basada en el Ciclo de Vida del Desarrollo Seguro de *Microsoft*, fue modificada para reflejar el cambio de perspectiva desde la del desarrollador al investigador externo. La metodología fue aplicada a la tecnología HVAC y posteriormente al dispositivo disponible, un enchufe inteligente, para dar un ejemplo de aplicación y arrancar con el proceso de análisis propiamente dicho. Es clave remarcar que la rápida evolución del mercado induce la necesidad de poner el foco sobre el proceso sobre el resultado, siendo clave la capacidad de actualizar las conclusiones rápidamente debido a la temprana caducidad de estas.

Tras concluir las pruebas sobre el dispositivo, se desarrolló una automatización de la explotación del mismo. Esto es prueba de la aplicabilidad, pese a algún punto de fricción, de la metodología y de la posibilidad de explotar el dispositivo. No obstante, estimar el impacto de dicha explotación requiere contexto externo, lo cual es tarea de proyectos futuros.

Abstract

The need for vulnerability assessment in HVAC (Heating Ventilation and Air Conditioning) devices arises from the risks that the introduction of smart capabilities into them creates. One such risk, and the one under consideration in this project, is the possibility to disrupt the electrical grid through coordinated attacks to poorly defended high-wattage smart systems. HVAC technology has seen quick development towards smart applications as evidence by the ample supply of products belonging to that category on marketplaces like *Amazon*.

The need to evaluate the risk in the European electrical ecosystem has led to the development of this project, whose main goal is to contribute to this effort by evaluating the vulnerability to this type of attack of products available in the European market. In order to achieve this in a systematic manner, a threat model methodology was produced to guide testing and applied to a relevant device.

The methodology proposed was based on *Microsoft's* Secure Development Lifecycle, but modified to reflect the change of perspective from that of the developer to that of a external tester. This methodology was applied to HVAC technology and to a smart plug device in order to provide an example and initiate the assessment proper. It is important to note that the rapid development of the market induces the need of focusing on the process rather than the outcome, because conclusions become old quick and adapting to new ones quickly is key.

After testing was done, an exploitation technique for the device tested was developed. This was proof of the soundness, despite a few points of friction, of the methodology and of the possibility of exploiting the device. However, assessing the impact of an exploitation requires external context and is left to further projects.

Just love the world
that won't
Love you back.
Pat Flynn, Have Heart

Acknowledgements

One night, not long ago, I went to sleep in tears convinced that this day would never come.

But that was before. I have overcome trials of body, mind and spirit. I have conquered this peak and I now bathe in the immense gratitude I feel to all of those that stood by me along this way. Thanks to the people around me, this path has been more bearable. I would not have made it this far without them.

Thanks to the doctors, nurses and health professionals whose work made me reach this point healthy and sane. Thanks to the people at ICAI for the professional excellence shown, it is a lesson I will not soon forget. Thanks to my friends for making this easier. Thanks to my family, specially parents, grandparents and in-laws for their immense support, tireless help and unending love. Thanks to Alicia for giving me the strength to keep on walking.

Special thanks to Grego for the late night reviews, the calls and messages that have made this work possible. It has been a pleasure working with you.

I will carry all of you in my heart.

José Antonio Font García
Madrid
July 12, 2023

Contents

List of Figures

List of Tables

Listings

Acronyms

AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
AP	Access Point
API	Application Programming Interface
C2	Command and Control
CoAP	Constrained Application Protocol
CRA	Cyber Resilience Act
DDS	Data Distribution Service
DNS	Domain Name Service
DNS-SD	DNS Service Discovery
DIY	Do It Yourself
EAP	Extensible Authentication Protocol
eFORT	Establishment of a FramewORk for transforming modern EPES into a more resilient, reliable and secure system all over its value chain
EPES	Electric Power and Energy System
GPT	Generative Pre-trained Transformer
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IBSS	Independent Basic Service Set
ICAI	Insitituto Católico de Artes e Industrias
IoT	Internet of Things
ITU	International Telecommunication Union
JNIC	Jornadas Nacionales de Investigación en Ciberseguridad
JSON	JavaScript Object Notation
MAC	Media Access Control
MaDIoT	Manipulation of Demand with IoT
mDNS	Multicast DNS
MobSF	Mobile Security Framework
MQTT	MQ Telemetry Transport
NIDS	Network Intrusion Detection System
OS	Operating System
OSI	Open Systems Interconnection

OTA	Over The Air
OWASP	Open-source Web Application Security Framework
PSK	Pre-shared key
SSID	Service Set Identifier
STA	Station
TCP	Transmission Control Protocol
TFM	Trabajo Fin de Máster
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	WiFi Protected Access
WPAN	Wireless Personal Access Network
WPS	WiFi Protected Setup
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introduction

Traditionally, the markets for home appliances and information technology were separate. However, developments in areas like microcontrollers and connectivity brought both worlds together. The term Internet of Things (IoT) dates back to 1999 but has expanded since then [1]. Although it is hard to define, institutions like The International Telecommunication Union (ITU) have given definitions that focus on global interconnection and enabling services [2]. This development has not only been a vast opportunity for businesses to digitize their chain of value but it has also brought a surge of connected consumer-end devices that go from smartwatches to locks, as evidenced by the ample offer that can be found in marketplaces such as *Amazon*.

However, there is not such thing as a free lunch, and a new paradigm such as IoT entails new risks. It is safe to say that a technology that has the potential to alter *things* in the world through the *Internet* has to be evaluated from a security standpoint. Moreover, consumer devices can collect personal information which is protected by the authorities in Europe [3]. There is research that demonstrate exploitation of smart consumer devices such as [4], but most relevant are attacks like *Mirai*, in which thousands of vulnerable IoT devices were employed as part of a botnet to attack the Internet through key services like *Dyn* [5].

In contrast, the development of the infrastructure and technology for electrical power and energy systems (EPES) is the product of a process spanning over a century under tight scrutiny from the authorities due to its strategic nature. This development has been fundamental for the development of the current world and it is critical in modern societies. In the European Union, electrical production is considered critical and is held to higher standards than normal industries [6].

Despite this special protection, motivated agents have been able to face increasing cybersecurity controls through time, resources and original ways to attack. Attacks in countries like Ukraine show [7]. The overlap of traditional EPES security with a new frontier of IoT is cause for concern in the European cybersecurity research space.

This document is a description of a project that is one step in addressing said concerns. The present section introduces the motivation of the project in the context that originated it and the project´s objectives. Next, it describes the methodology employed and the resources available. Lastly, this section describes the structure of the document.

1.1 Motivation

This work was developed under the eFORT (Establishment of a FramewORk for transforming modern EPES into a more resilient, reliable and secure system all over its value chain) project, which is an European project that brings different entities together under one purpose: defending Europe's EPES [8]. In order to achieve this, an assessment of threats regarding said infrastructures is needed. One such threat arises from the overlap between EPES and consumer grade IoT high-power devices.

A series of works presented in the USENIX conference discussed a new typology of attack, a manipulation of demand through IoT devices (MadIoT) [9], [10], [11]. This attack, similarly to the famous botnet attack malware Mirai [5], gathers a collection of devices to attack its target. However, instead of Internet services, targets in this scenario are EPES and the members of the botnet are smart high wattage devices rather than cameras and routers. The original works assume exploitability of the devices, which is what this project tried to assess.

There are several different technologies in the European market that can entail a risk through this attack scenario, such as solar technology and electrical vehicle charging stations. One such group of technologies, HVAC (Heating, Ventilation and Air Conditioning), is relevant for the following reasons: (i) HVAC systems are considered high-wattage as their consumption can reach the thousands, (ii) they are fairly ubiquitous in Europe and (iii) IoT solutions to transform a traditional (i.e. not connected) installation into a smart one are becoming more common, as evidenced by products such as the *Tado* Thermostat, belonging to a product family with an user review volume in the thousands. In order to assess the risk that this technology contributes to EPES in Europe, a systematical vulnerability analysis of relevant devices is needed.

1.2 Objectives

The main goal of this project consisted on meeting this need by conducting an assessment of vulnerabilities in a relevant device. To achieve this in a systematic manner, a threat model for the relevant technology was needed to guide the testing along with a study of devices in the market. Once the model and the market survey had been made, a specific threat model for the device was built in order to define test to be performed. Lastly, the tests were performed and conclusions were drawn. The goals for the project were:

- Defining a threat model for the technology.
- Applying the threat model for the device.
- Testing the device for vulnerabilities following the model.
- Extracting conclusions to help reach decisions regarding EPES' security.

1.3 Methodology

In this section, the methodology and planning employed will be discussed. The methodology for the project has been based on deliverables and the most relevant information for said deliverables has been included in the present document. It is important to note that the relevant documents were open and updated if findings along the development of further steps were significant. The next paragraphs detail the different stages during the course of the project and the deliverables produced during said stages

The first stage of the project consisted on analysing the state of the art in order to define the project's context, derive the scope and gain relevant knowledge needed to carry out the testing. The main deliverables for this stage were a state of the art analysis, a market survey and initial testing guidelines.

After the theoretical ground was established, work on the threat modelling and the testing started. The output from this stage was a threat model methodology proposal that was presented during JNIC 2023 and installation guides for the target device, the Sonoff Mini R2.

Next, the threat model methodology was applied to the device and testing guides were produced. During this stage, tests were carried on the device and findings and evidence were collected

Lastly, the last stage of the project consist on developing the present document as well as producing useful tools for the next projects derived from this one.

For the scheduling of the project, the timeline has been divided into weeks, from 01/23/2023 as a reference starting date to 06/31/2023 as a finish date, resulting in 23 weeks to allocate the different tasks described above. The structure of the tasks along with the proposed schedule is shown in the Gantt chart in figure 1.1.

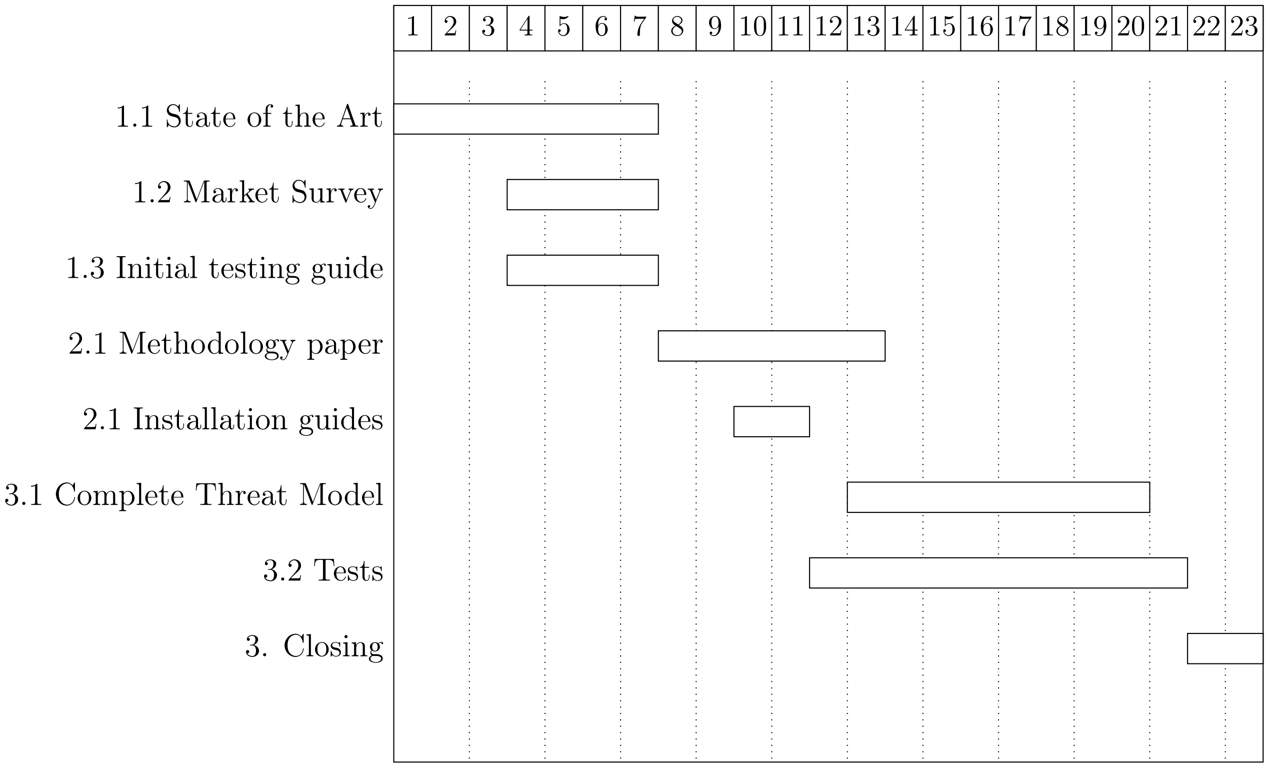


Figure 1.1: Gantt chart for the project

1.4 Resources

This section covers the resources employed to carry out the testing. The first part is a description of the hardware resources employed, followed by a description of the software resources.

1.4.1 Hardware resources

WiFi Pineapple

Figure 1.2: WiFi Pineapple

The WiFi Pineapple is a tool for WiFi auditing that is formed by three WiFi interfaces controlled by a Linux OS (Operating System) that allows, among other things, WiFi monitoring and discovery, deploying a controlled access point (AP), establishing traffic rules and executing custom code.

The device employed during testing was the *MARK VII BASIC* and was used for sniffing network traffic and subverting network functions such as DNS (Domain Name System) replies, though it offers more capabilities that were not leveraged.

Linksys Router

In order to replicate a common home installation, an off-the-shelf router was employed. In this case, the tool of choice was the *Linksys E5400* router. Though its role could arguably be covered by the aforementioned WiFi Pineapple, it was employed due to the realistic emulation of a home environment and ease of installation.

Android Phone - Xiaomi Mi Phone 5

An Android phone was employed to run mobile software through real network interfaces. Although the use of emulation was considered, the employment of real hardware proved to offer the least resistance for testing.

1.4.2 Software resources

tcpdump and libpcap

In order to capture and filter traffic *tcpdump* and *libpcap* were employed. They are a set of command line tools for traffic analysis. During the project, they were employed to capture traffic using simple filters on the WiFi Pineapple.

Wireshark

Wireshark is an open source network protocol analyzer that offers a graphical user interface along with a myriad of tools to inspect network traffic including filtering, live capture, protocol decoding and exporting. Its main use during the project was analyzing traffic captures extracted during testing in order to use its packet inspection tools to extract useful information. The version employed during testing was version 4.0.5.

Postman

Postman is a tool for testing APIs (Application Programming Interface) in a team. It allows, among other things, the creation of work spaces in which define API requests, launching them and receiving responses. Its uses during testing were testing the device's accessible JSON (JavaScript Object Notation) API systematically.

MobSF

MobSF, or *Mobile Security Framework*, is a mobile application framework to perform automated testing. It offers a wide variety of capabilities including static and dynamic analysis, continuous integration and malware analysis. During the project, it was used for testing the Android application provided by the device's seller for control.

Python

Python is a programming language that allows the quick creation of scripts. The main motivation for its use it the number of network and cybersecurity libraries available. During testing, it was employed to automate attacks and produce a working proof of concept script to automatically attack the device.

Generative Pre-trained Transformer tools

Generative Pre-trained Transformers (GPT) [12] are a methodology to train large artificial intelligence language models. During the development of the exploit, ChatGPT was used to help fix errors in the code and accelerate development. Additionally, it has been employed during the writing of this document to help with formatting LaTeXcode.

1.5 Document structure

After the context of the project has been introduced the document continues with an overview of the state of the art. Then, a vulnerability analysis is introduced with a description of the threat model methodology employed. Next, the threat model defined for the device is explained along with the testing performed. Lastly, the document ends with an overview of the findings and a proposal for different options for continuing with the work done.

Chapter 2

State of the Art

The following chapter covers the state of the art for the technologies, methodologies and areas of knowledge related to the project.

2.1 Internet of Things

The term IoT has been defined by the ITU-T as a "global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [2]. The key definition of "thing" implies identification and communication. These enabled services translate into many applications for industries, governments and consumers. A proof for the latter is the wide availability of products that fall under this definition in popular marketplaces such as *Amazon*.

IoT has had a major effect on the technological environment and is expected to grow over the next decade, as evidenced by [13]. This creates many opportunities for different players, making the environment reliant on a number of different technical solutions that are constantly evolving. The heterogeneity and dynamism of the environment are also mentioned in the ITU definition. This entails the need to overview the most relevant of these solutions for the project at hand.

Another element mentioned in the definition, the most relevant for this project, is the need for security. The inclusion of new *things* in the *Internet* creates a new attack surface, increasing both the probability of threats to take place and their impact.

2.1.1 Technologies

The environment for IoT technological solutions is vast, as it contains a wide variety of providers for a big number of applications. Hardware solutions vary from microcontrollers to traditional motherboards running all sorts of firmware and operating systems. Regarding communications, there are several standards on all layers of the OSI model, ranging from Ethernet to HTTPS.

Regarding this project, the most relevant technologies are those commonly found in consumer applications. When it comes to local connection and networks, the most common solution for consumers is WiFi for Internet access, along with Bluetooth (or BLE) for communication between devices with little configuration. Additional technologies such as ZigBee or LoRaWan exist, but no devices employing these technologies were within the scope of the project. Additionally, the device can implement physical and virtual interfaces, with the latter being the most relevant for the project. These can be implemented as a web service or a smartphone app. The application layer communication can be implemented over HTTP (or its secure version, HTTPS) or other protocols such as MQTT, CoAP, XMPP, AMQP or DDS [14]. Lastly, additional technologies have been found during the course of the project, such as DNS, and are reviewed in this section.

WiFi

The most common technologies for WLAN (Wireless Local Area Network) deployment are those of the 802.11 family [15], commonly known as WiFi. On the most generic level, it implements MAC level networking and is comprised of APs that define the network and stations (STAs) that partake in it. The most common application in homes is the infrastructure mode, in which the AP acts as the central element and handles all traffic, though independent mode (or IBSS) does exist, in which STAs communicate directly without the need for an AP. Regarding security for the scope of this project, the most relevant features of WiFi are encryption to protect the channel, authentication, network availability and device discovery. Regarding encryption, the standard has experienced a development from open APs to protection through WPA3. Regarding authentication, WiFi provides several options from static keys and open networks to extended mechanisms that implement certificates, user passwords, etc. Availability compromise and device discovery are somewhat inherent to the wireless nature of WiFi, though mitigations might exist such as MAC filtering and network intrusion detection systems (NIDS).

The existing encryption methods are:

- **Wired Equivalent Privacy:** WEP was the first implementation of WiFi encryption. The algorithm works by using the exclusive OR operation between the plaintext, comprised of the message and a checksum, and the pseudorandom output of applying the RC4 algorithm to the key and an initialization vector [16]. In the standard definition, 40-bit keys were implemented, though they could be extended to 104 bits. Today, it is considered insecure and its use is not recommended as evidenced by works such as the aforementioned [16].

- **WiFi Protected Access:** WPA improves on WEP by increasing both initialization vector and key to 48 and 128 respectively. Additionally, it improves key management through the temporal key integrity protocol, implements better message integrity through integrity checks and prevents replay attacks by implementing a frame counter. It was quickly deprecated in favor of WPA2 [17].

- **WPA2:** WPA2 improves security by implementing AES into encryption. It is the standard and the most widely used. However, it was proven vulnerable in [18].

- **WPA3:** WPA3 implements many security improvements like per-device encryption, secure key exchange and improved cryptographic algorithms. This implementation, however, is not as extended and vulnerabilities do exist [19].

The relevant authentication methods are:

- **No authentication.**

- **PSK authentication:** The AP defines a key that is needed for establishing connection and to generate encryption keys. It is widely used in homes. Additionally, users can employ WPS to connect devices to the network using a shorter PIN or by pressing a button on the AP, though this practice is not recommended.

- **802.1X/EAP:** The most used in enterprise networks. On it most basic implementation, the station is authenticated through the use of an authentication server like RADIUS. Its functionality can be expanded with EAP.

Some additional security features are:

- **MAC Filtering:** Current consumer grade WiFi equipment can implement features for allowing or disallowing connections or traffic at data link layer. Allow-lists apply filtering by specifying which addresses are permitted and are considered more secure than deny-lists, which in turn define prohibited users. One key note is that MAC addresses can be dynamic or falsified.

- **Firewalls:** A firewall is a cybersecurity element that allows or blocks traffic in order to prevent malicious actions. They are one of the methods of implementing perimeter control. Depending of the layer at which it operates, a firewall can filter traffic through MAC addresses, IP and Port, application type or packet content. The traditional implementation of firewalls is rules based. The home router employed implements denial of service protection (it is not clear how it achieves this), multicast and anonymous request filtering, redirect filtering, port 113 filtering and different application layer filters, such as proxies or cookies.

- **NIDS:** A network intrusion detection system is a cybersecurity element that identifies potentially dangerous behavior in the network. In order to achieve this, a NIDS scans traffic to search for evidence of abnormal or suspicious behavior, such as attack signatures or anomalous protocols.

HTTP and HTTPS

HTTP is an application protocol for the TCP/IP stack whose main purpose is transmitting files, specially formatted text files like HTML, CSS or JavaScript files. It implements an architecture in which a server (on TCP port 80 by default) replies to requests made by clients. The client can characterize the request through metadata, in the request headers, or by parameters. Additionally, the client can employ a variety of methods for the request, the most common being GET for sending simple requests with parameters in the URL, or POST for sending more elaborate requests with a body containing information, like a JSON object.

However, HTTP implements no security. HTTPS was born to fill this void. HTTPS uses cryptographic protocols to verify server identity through certificates, secure symmetric key exchange and channel encryption. There are two different sets of protocols in use, SSL and TLS. The former, however, was proven insecure and its use is not recommended.

MQTT

MQTT is a lightweight protocol that offers a viable alternative to HTTP in IoT applications. A simple implementation consists of clients that exchange publish/-subscribe messages with a broker regarding a specific topic. The protocol is built on top of the TCP/IP stack. It allows authentication but does not provide encryption by default.

Following the example oh HTTP, MQTT can work with SSL/TLS to create an encrypted channel and authenticate the server through certificates.

DNS

DNS is an essential part of the Internet whose purpose is converting human readable domain names into numeric IP addresses. In order to achieve this, devices who need domain name resolution query DNS providers known as name servers. DNS servers implement a decentralized hierarchy that allows the scalable search of matching domain and IP records.

However, different types of attack leverage this functionality to create channel through which exfiltrate data or to redirect users to fraudulent domains. In this project, DNS attacks have been performed in order to try to impersonate the cloud service with which the device interacts.

Additionally, Multicast DNS (mDNS) is a protocol for resolving hostnames within multicast domains without the need for a name server. In order to achieve this, queries are sent to a relevant domain who are answered by the relevant host subscribed to that domain. Alternatively, DNS-SD (Service Discovery) solves the search for services available by implementing service registering on the local domain and service queries.

2.1.2 Security

In the market for IoT for consumers, security can be in a contradiction with functionality. The motivation for reduced prices, newer functionalities and low time to market can clash with the extended requirements and work that security entails. Moreover, while there are brands that market themselves on higher security requirement, they usually hang on the higher end of the price spectrum, leaving consumers with lower budgets exposed. Additionally, updating devices to keep up with cybersecurity development is problematic in some cases, as is reflected on the cyber resilience act (CRA) of 2022 [20].

The following paragraphs go over the relevant IoT attacks for the project (MadIoT and Mirai), the OWASP standard for IoT and the recent CRA.

MadIoT and Mirai

The motivation for this project starts with the definition of the MadIoT attack in 2018 [9], advancing work by A. Dabrowski [21] the year before. In this paper, S. Soltan theorizes about a new form of attack, labeled MadIoT, in which an attacker would exploit oversights in smart high-wattage devices to attack EPES. A coordinated alteration of power consumed, which could reach the order of gigawatts, which could affect frequency operation, cause line failures and blackouts or increase energy prices.

The next year, B. Huang [10] presented possible protections against this attack, arguing that the attack would not be as catastrophic as previously thought. However, T. Shekari [11] proposed an advanced attacks by coordinating attacks to weaker locations in the grid, rather than attacking randomly.

Regarding the exploitability of devices, the events of the Mirai attack [22] show that this kind of attack is a reasonable concern. During the attack, over $600,000$ bots were employed to attack the American internet infrastructure.

OWASP

The page for the OWASP project regarding IoT [23] does not contain much information yet. However, the top 10 vulnerabilities proposed are relevant to the project:

1. Weak, guessable, or hardcoded Passwords.
2. Insecure network services.
3. Insecure Ecosystem Interfaces.
4. Lack of secure update mechanism.
5. Use of insecure or outdated components.
6. Insufficient privacy protection.
7. Insecure data transfer and storage.
8. Lack of device management.
9. Insecure default settings.
10. Lack of Physical Hardening.

However, some vulnerabilities, like the lack of physical hardening or lack or device management, are not relevant to the attack scenario at hand.

Cyber resilience act

Recently, the European Comission released this proposal [20] with the purpose of improving security in the European Union. This proposal, whose vote is scheduled for *July 19th*, proposes increased requirements digital products, a classification which includes IoT. The proposal includes sanctions of up to 15M€, security requirements and the constitution of vigilance authorities.

2.1.3 HVAC

Different models for IoT implementations have been revised. The most relevant for the project was the work by J. Serra [24] in which the smart HVAC control is broken into three subcomponents: actuators, sensors and control. Other relevant information is part of the market survey described later in the document.

Chapter 3

Vulnerability analysis

3.1 Market survey for Smart HVAC systems

The market for smart consumer grade IoT devices for the home has grown from nonexistent to a thriving one in a short span of years and there is no end to this trend in sight. Applications range from controlling lights to taking real time measurements of elaborate systems such as gardens and deploying automated actions. The user experience has evolved rapidly from being a niche for tech savvy enthusiast to out-of-the-box systems, controlled by established smartphone apps or even smart home assistants, such as *Alexa* (2014).

Typical Smart HVAC implementations offer more than mere control. Some include features like consumption control, scheduling, automated tasks or voice assistant support.

While modern installations can include digitization elements, like this system by Samsung, older installations do not support smart control capabilities natively.

However, a wide variety of products can allow older equipment to achieve the desired functionality. One common way to achieve this are IR controllers that can be installed with the air conditioning unit in sight and connected to WiFi to act as a relay between user and unit. This implementation, however, only works with IR equipment and that is not the case for common water heaters or typical heating systems.

A different strategy to achieve digitization is through the use of smartplugs that are placed between the load and the power supply and can be used to detect current draw and control it. However, some smartplugs may require increased technical knowledge due to the need for connectivity through stripped wires rather than sockets.

Other devices can be directly connected to the equipment, either directly inside the unit or through thermostat wiring. These require technical knowledge for installation but offer the most capabilities.

A summary of consulted devices is available on tab. 3.1. The reviewed information shows that there is a significant number of customers for the consulted devices, willing to incur in varying degrees of expenditure, from under twenty euro to thousands. The disparity in prices hints to a possible disparity in security, due to the motivation to cut costs in order to bring fast and cheap products to the market.

Table 3.1: Devices Surveyed

Device	Price	Power	Reviews	Type
Samsung RAC, R32, Mural gama Wind Free Elite	> 1000€	$2.5kW$	-	Integrated
Btcino XW8002E	> 100€	Any	91	Thermostat
Tado^{o} Termostato	> 100€	Any	1429	Thermostat
Airzone-Aidoo	> 100€	Any	86	Installable
Broadlink RM4	< 50€	Any	136	IR
Sensibo Air	> 100€	Any	721	IR
Tado^{o} Mando	> 100€	Any	11	IR
Sonoff Mini R2	> 10€	$2.2kW$	2342	Smartplug
GreenBlue GB109G	< 100€	$3.6kW$	6	Smartplug

A summary of consulted devices is available on tab. 3.1. The reviewed information shows that there is a significant number of customers for the consulted devices, willing to incur in varying degrees of expenditure, from under twenty euro to thousands. The disparity in prices hints to a possible disparity in security, due to the motivation to cut costs in order to bring fast and cheap products to the market.

Lastly, there is a highly developed ecosystem for Smart Home applications, allowing the extension of individual devices' capabilities with the interconnection of other devices, software and cloud resources. Additionally, there are several voice assistants on the market that can be integrated with most reviewed devices. Some of the aforementioned ecosystem apps are:

- Google Assistant.
- Amazon's Alexa.
- SmartThings.
- Philips Hue.
- Home Assistant.

- IFTTT.

3.2 Threat Modelling methodology

The following section will review other threat models employed in similar scenarios, describe the process for the threat modelling methodology proposed and provide the argumentation for the choices that were taken in order to produce it.

First, the HEAVENS model was reviewed. The main reasons for its inclusion were its focus on automotive electrical and electronic systems [25] and its relation with the European cybersecurity space. It is, however, rather cumbersome for the purpose of this project.

Next, a methodology developed by Microsoft around the STRIDE taxonomy has been reviewed. This methodology can be broken down into nine phases [26]. However, an article also published by *Microsoft* proposes a simplified process consisting of 4 steps: decompose the system into relevant components, analyze each component for susceptibility to the threats, mitigate the threats and repeat [27].

The taxonomy that provides the name, STRIDE, defines categories for classifying threats: spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege [27]. As stated previously, the first step in the process is defining the functional components of the target system. For this purpose,a Data Flow Diagram (DFD) is employed, built from the following categories: external entities (rectangles), data flows (arrows), processes (circles for simple processes and double circles for complex ones) and data stores (two parallel lines; plus an additional element relevant to security being trust boundaries [27] (dotted line). Next, the model's categories, developed using threat trees, are used to propose mitigations which are then prioritized and implemented. After mitigations are performed, the process is iterated.

This methodology has been chosen as a base to build the model for its simplicity without compromising usefulness along with the wide availability of documentation due to its widespread use. However, it focuses on secure development rather than vulnerability assessment. For this purpose some elements have been added, detailed in the following paragraphs.

The key difference is the separation of the model into two blocks, one to provide a general model for the technology and a second to develop the generic model for a specific device, system or implementation. The reason for this is to provide a basis to quickly develop assessments for new devices or systems that appear in the market.

The purpose of the first block consists on defining a description of the technology from a cybersecurity perspective. In order to achieve this purpose, the proposed steps for this block are:

1. Define attack scenario: This helps situate the technology in regards to the power infrastructure. For this project, the attack scenario is of a MaDIoT attack.

2. Define generic DFD of the technology and establish threats using the STRIDE taxonomy: This step defines a baseline DFD for the technology to use as a guide for specific device. It is also recommended to provide a list of relevant technologies used in the market for devices in the relevant group.

3. Establish weaknesses: Once the DFD is developed, it can be used to provide with a list of weaknesses that are relevant for the attack scenario.

Next, the second block aims to develop the contents of the first block for a specific device in order to provide an analyst with a list of controls that are relevant to the device:

1. Review external security notes and define assumptions for a device: reviewing security notes for the device provides with a clear understanding of the operation of the device which is in turn condensed in a list of assumptions.

2. Specify generic DFD for a device: Once specific information pertaining the device has been reviewed, an specific DFD can be made, containing the specific elements from the generic model that apply.

3. Define control list: With the information defined in the previous steps, an analyst is able to produce a list of mitigations to check on the device.

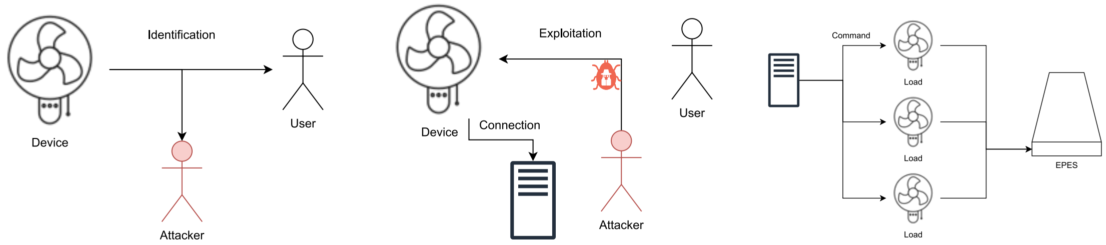
After the process has been completed, a list of controls to be tested is available, along with the documentation to support the selection of said controls. This is useful for the assessment of the risk that particular devices of a specific technology entail for a given infrastructure.

The following sections are dedicated to explain how this methodology has been applied to HVAC, electrical vehicle charging and photovoltaic technologies in order to systematically assess the risk they suppose to the power infrastructure.

3.3 HVAC Generic Threat Model

The following section will constitute an example of a development of the threat model as it was performed during the assessment of the susceptibility of HVAC systems to be targeted by a MaDIoT attack.

Smart HVAC system's security is relevant to risks in power grids. This is because quick developments in smart solutions are gaining an increasingly relevant



(a) Discovery (b) Exploitation (c) Attack

Figure 3.1: Attack Scenario

position in the market for home HVAC systems, as evidenced by the ample supply of relevant devices in marketplaces such as Amazon, and its share of the power consumed in homes [28]. While installation of HVAC units requires certification, adding customer devices such as smart control systems does not, which makes HVAC a relevant object for analysis.

The first step in the methodology consists of defining the attack scenario. From the information provided previously, it can be derived that the scenario in scope is that of a MaDIoT attack. This scenario defines that the goal of the attacker is to alter the power being consumed by the user, which removes the focus from other possible impacts, such as using the system for a lateral movement in the network. In this scenario, an attacker would be placed outside of the users' home network and gain access to the operation of the system by being able to manipulate the target parameters of the device (i.e. temperature, humidity, air quality, etc.) or turn it on or off via direct command or schedule.

In order to attack the device, the device should be identifiable. Different implementations could exist, and some could offer more information or have further reach than others. For example, Bluetooth broadcast would offer potentially dangerous information but an attacker would need to be within range, which is not compatible with the attack scenario. After identification, the attacker would have to deploy some form of schedule for the attack or a way to relay commands to the device, which could be achieved by directly deploying malware to the device or by gaining control of another element within reach of the device. Lastly, the attacker would use the remote C2 station to launch the attack to several targets. This is shown in fig. 3.1.

The next step according to the methodology is defining the generic DFD for the technology. The diagram in Fig. 3.2 reflects, in generic terms, the interactions of a smart HVAC installation with the external elements it requires to work. First, the network defines its own boundary of trust as it marks what is local to the installation and relays connections with remote elements. The most common technology for this purpose is *Wifi*, although others can exists such as *Bluetooth*

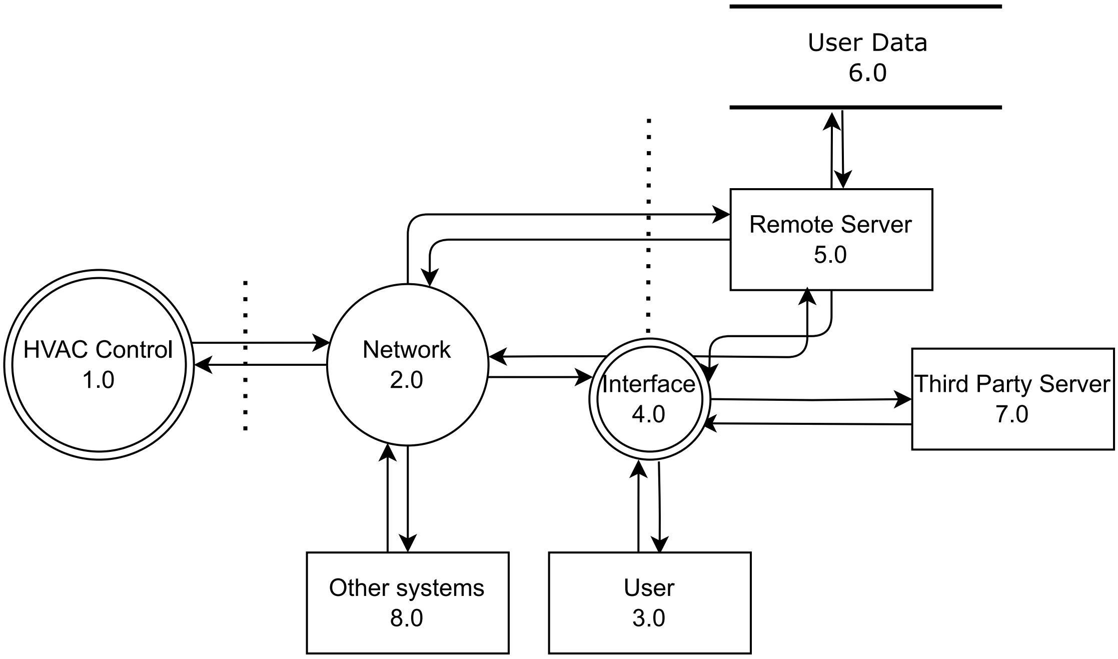


Figure 3.2: Network DFD

or *Zigbee*. Next, the user and the user's interface is placed on the boundary, as it can act as a local agent and a remote one. This interaction happens within a specific device, such as a smartphone or a computer. Additionally, there can be a remote server handling user data and interacting with the system.

The DFD in Fig. 3.3 reflects, on the other hand, the elements that can conform a smart HVAC system. Such a system can include a series of sensors that direct information to a control unit that will, in turn, communicate that information to the user and send commands to the actuators in order to affect the ambient. The key takeaways from this diagram are the different paths commands can flow through the control unit (either directly or through the network) and the possibility for the control unit to keep an activity log.

Additionally, the diagram in Fig. 3.3 defines the functional elements for an interface with the system. It is based on a central process that manipulates, the operating system, that handles application data, user interaction, other software in the device and networking, which can occur through different interfaces as is the case for smartphones.

Lastly, with the DFD developed, in can be used to define a list of weaknesses using the STRIDE taxonomy. The taxonomy categories are as follow:

- Spoofing: Authenticity.
- Tampering: Integrity

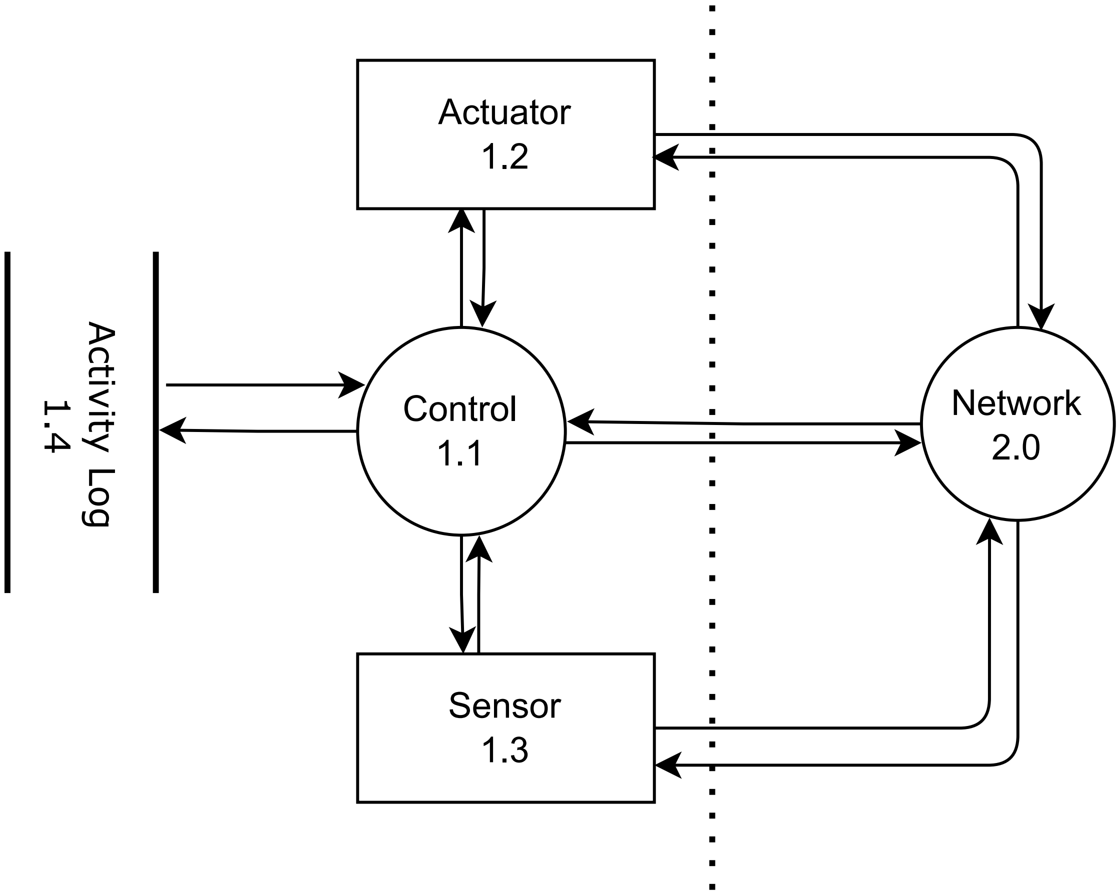


Figure 3.3: System DFD

- Repudiation: Non-repudiability. Only file are Logs.
- Information disclosure: Confidentiality
- Denial of Service: Availability
- Elevation of Privilege: Authorization

The last step of the methodology for defining a generic threat model consists on employing the conclusions drawn from 3.2 to establish weaknesses.

Going back to the attack scenario, the first step for an attack would be identification. This depends greatly on the network employed and the implementation. Some networks may have a farther reach, like IP based networks, while others would make an attack very difficult due to reach, like Bluetooth. Additionally, some implementations could be based on stronger authentication, leaking less

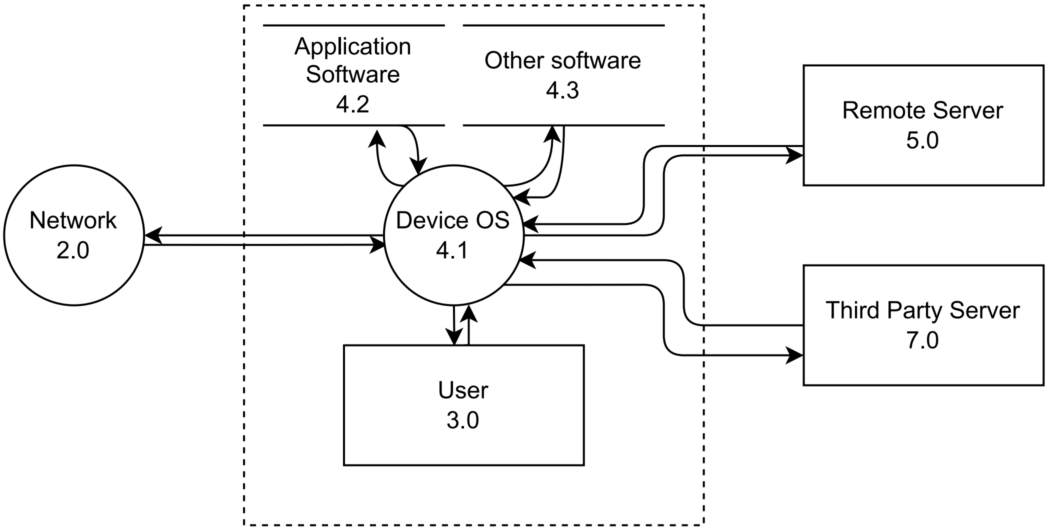


Figure 3.4: Interface DFD

identification during operation. In order to establish a connection between device findings and exploit type, an attacker could perform heuristics on the leaked information. This is shown on fig.3.5.

During exploitation, an attacker could exploit vulnerabilities in the device´s design to gain control or execute unwanted functionalities. Some devices can be connected to the internet while others can only operate in local networks. However, these last type of devices are not secure since other elements in the network could be exploited and be employed to relay traffic from the attacker. The goal of an exploit is to alter the state of operation of the device: either modifying the power consumption or turning it on/off completely. One route to achieve this would be attacking authorization in order to gain the necessary capabilities to send commands. An alternative route would be temporarily disabling the device and impeding its consumption. Additionally, an attacker could improve the effectiveness of the attack by disabling legitimate commands and disabling logging capabilities.

Lastly, the attack itself would require a steady connection between the exploited device and the attacker's C2 (Command and Control) infrastructure. Alternatively, devices with increased capabilities could be scheduled during the exploitation phase to perform the attacks independently.

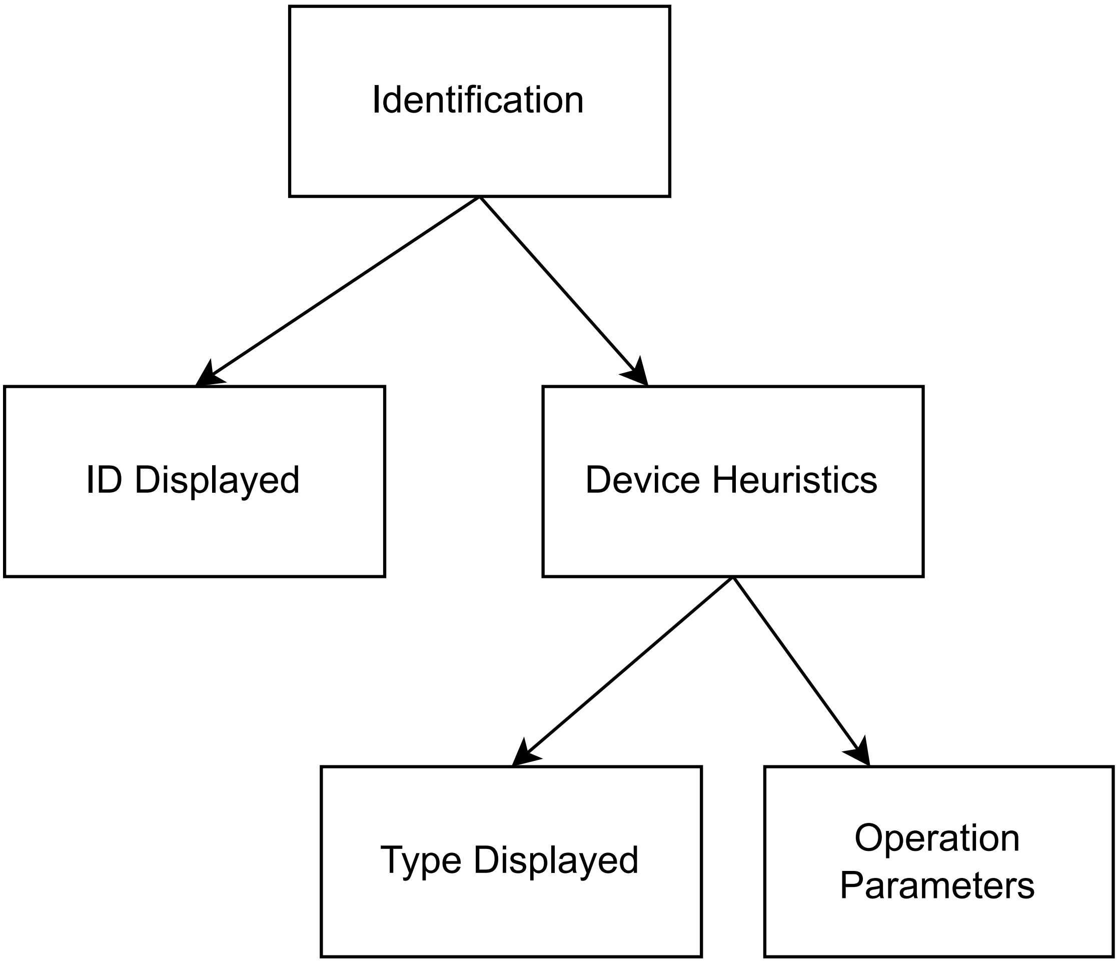


Figure 3.5: Identification threat tree

Table 3.2: Threats table

Element type	Name	Number	Threats
External Entities	User	3.0	S,R
	Remote Server	5.0	S,R
	Actuator	1.2	S,R
	Sensor	1.3	S,R
	Third Party Server	7.0	S,R
	Other Systems	8.0	S,R
Processes	Control	1.1	S,T,R,I,D,E
	Network	2.0	S,T,R,I,D,E
	Device OS	4.1	S,T,R,I,D,E
Data Stores	Activity Log	1.4	T,R*,I,D
	User Data	6.0	T,I,D
	Application Software	4.2	T,I,D
	Other Software	4.3	T,I,D
Data Flows	User interaction	(3.0↔4.1)	T,I,D
	Remote Network	(4.1↔5.0)	T,I,D
		(4.1↔7.0)	T,I,D
	Local Network	(4.1↔2.0)	T,I,D
	Other Systems	(8.0↔2.0)	T,I,D
	Remote Interaction	(5.0↔2.0)	T,I,D
	Direct Control	(1.1→1.2)	T,I,D
		(1.1→1.3)	T,I,D
	Direct Report	(1.3→1.1)	T,I,D
		(1.2→1.1)	T,I,D
	Indirect Control	(1.1→2.0)	T,I,D
	Indirect Report	(1.3→2.0)	T,I,D
		(1.2→2.0)	T,I,D
	Control Relaying	(2.0→1.2)	T,I,D
		(2.0→1.3)	T,I,D
	Report Relaying	(2.0→1.1)	T,I,D
	Logging	(1.1↔1.4)	T,I,D
	Execution	(4.1↔4.2)	T,I,D
		(4.1↔4.3)	T,I,D

Chapter 4

Sonoff Mini R2 Assessment

This section describes the relevant details about the device in regards to its contribution to risk in the EPES environment. In order to achieve this, the most important information about the device is provided in the following paragraphs, along with commercial information relevant to the potential impact. Next, a section detailing the particular threat model developed for the device describes all relevant information available for the device. With that information, a series of relevant controls have been defined for the purpose of testing, which yielded results that are explained afterwards. Lastly, an overview of the testing is provided.

The Sonoff Mini R2 is a device designed by Sonoff. It is a smart switch device that is connected between a power source and a power load to switch it on and off, while also allowing for a physical switch to be connected to operate the device manually. The main operation mode its through the manufacturer app, that is able to connect to eWeLink. This app allows the connection to third party interfaces such as *Alexa*. Alternatively, the user can access a Do it Yourself mode, or DIY, in which the device is configured by the user directly without the app. The device allows the setting of timers to schedule repeated activity and flashing new firmware through WLAN. Additionally, the app allows to create scenes in which run different devices together and allows the delegation of control to other users through its cloud. The device installation is aimed at contractors or niche users as it is meant to be installed along with the rest of the electrical installation in the walls.

Regarding potential impact, the device is certified for a consumption of up to $2kW$, which falls under the assumptions seen in [9] for high power. It is important to note that the impact is limited as this is the *maximum*, which means that most applications do not reach that level and therefore the impact is limited. On the other hand, there is no estimation available about how many devices are deployed in the European Union. However, the page on the Spanish Amazon store shows over $2,000$ reviews.

4.1 Threat Model

4.1.1 Documentation Review and Assumptions

Sonoff provides three brief documents on the product's webpage: specifications[29], a user manual[30] and a quick start guide [31]. Additionally, there is a section for articles regarding several aspects of the company's products such as tutorials, news or other kinds of information. One such article of relevance is the DIY API mode guide [32]. Lastly, there is information to be found on resources provided by ITEAD, a parent company to Sonoff, such as a GitHub open source project for the DIY mode [33].

The basic documentation provides information about the installation process for the device and key details about the device such as functionalities or encryption that can help guide the assessment process. The device offers two connection methods for the app control mode, a quick pairing mode in which the device is able to search for the device; and a compatible mode in which the user has more interaction to specify the device. The device's functionality include: remote control, timing, voice control, share control, smart scene, WLAN control and status sync. Additional information is listed in table 4.1.

WiFi	IEEE 802.11 b/g/n 2.4GHz
Security	WPA/WPA2
Encryption	WEP/TKIP/AES AES-128-CBC/PCKS7Padding Key is MD5 hash (16B) of API Key
MCU	ESP8285
Certification	CE/FCC/ROHS
Default AP Password	12345678

Table 4.1: Documentation information

The external documentation provides useful information about the operation of the device in the DIY mode. Even if this mode is not the central focus of the assessment, it is reasonable to assume that the control through the app is similar, though stricter configuration defaults may be applied. To address this issue, findings derived from the information available on the DIY mode should be validated on the default mode. The documentation describes two important ways to interact with the device: a mDNS/DNS-SD service for finding information about the device, and a RESTful API to access functionality.

Regarding the discovery service, the documentation describes two approaches: a default one through mDNS, and a compatibility one based on DNS-SD. The device will make the same information available through either method, though the first one is more straightforward. The following information can be found:

- IP Address.
- Hostname: formatted as *"eWeLink_[DeviceID]"*, which helps identifying the device as a part of the eWeLink family of products and the specific *"DeviceID"*, which is useful for interaction with the API.
- Other default identifiers: Such as using the service type *"eWeLink.tcp"* or using *"txtversion=1"*.
- The API Version.
- The *"seq"* parameter, which is a counter for petitions made.
- Information on the device's state: Though it can be encrypted, the query response will include a JSON object with the following additional information:
    - Switch status.
    - Startup status and other functional information such as pulse status.
    - SSID.
    - *"otaUnlock"* status: a boolean parameter that controls whether the firmware can be flashed.

Regarding the RESTful API, the documentation provides information about the different inputs it can handle, such as URL paths and parameters, and the outputs it can provide, such as requested data or error codes. The URL paths are endpoints for the different functionalities that the device offers. All of them stem from the basic *"/zeroconf"* route and are served through port 8081 by default. Paths can be consulted on table 4.2. On the other hand, JSON parameters and, though there are some generic parameters, they can depend on the specific functionality accessed, in which case they are put into an object called *"data"*. They work similarly both in request messages and in responses and can be found in table 4.3. Lastly, error messages appear in responses were there was something wrong and can help the assessment in guessing what is happening inside the device. They can be found in table 4.4.

The following assumptions were made:

- The attacker is outside the network. However, local components could be accessible to an attacker.

Path	Description
switch	Turn the device on and off
startup	What happens when the device power supply turns on
signal_strength	A negative integer in dBm
pulse	Activation of the inching function
wifi	Change WiFi connection
ota_unlock	Disengage the block on firmware updating
ota_flash	Download new firmware over HTTP
info	Get device information

Table 4.2: Paths in the documentation

Parameter	Functionality	Description
deviceid	General	The identifier for the device
sequence	General	Serial number of the request
selfApiKey	General	User key for the cloud API
encrypt	General	Boolean for encryption, data will be Base64 encoded when true
iv	General	Initialization Vector for encryption, Base64 encoded
seq	Response	Interactions counter
error	Response	Error code
switch	switch	Turn the device on/off
startup	startup	When the device is powered on, the output is set to on/off/stay
seq	signal_strength response	Received signal in dBm
pulse	pulse	Turn the inching mode on/off
pulseWidth	pulse	Milliseconds. Multiple of 500
ssid	wifi	Target WiFI SSID
password	wifi	Target WiFi password
downloadURL	ota_flash	URL to download firmwware
sha256sum	ota_flash	Integrity check for download
bssid	info response	Unique identifier for the host WiFi
fwVersion	info response	Current running Firmware

Table 4.3: Parameters in the documentation

Code	Description
0	Success
400	Incorrect format
401	Encryption required
404	Not supported deviceid
422	Invalid parameter
403	OTA update: OTA is locked
408	OTA update: Pre-download timeout
424	OTA update: URL unreachable
471	OTA update: Failed integrity check
500	Error during OTA unlock: Unsupported API key or device ID
503	OTA unlock unavailable

Table 4.4: Error codes in the documentation

- The target mode of operation is the default utilization with the *eWeLink* app. However, the DIY mode and its documentation has proven useful to discover the inner workings of the device. This is the product of a deeper assumption that both modes use similar code with different configurations.

- The device is vulnerable to inherent faults of WiFi, such as channel confidentiality and integrity compromise through key reuse and denial of service through de-authorization.

- Although the pairing process may be vulnerable, it is not within the scope of the attack scenario, as it would require advanced interaction that is not coherent with a botnet attack. Nonetheless, information extracted during pairing could help further tests.

4.1.2 Data Flow Diagram

Regarding the DFD there are three levels of control that determine the reach of the testing. First, the most relevant group, would be items within control of the manufacturer. These are the items that are directly within the scope of testing. The second group is conformed of items whose security is controlled by the user. Those are relevant to security, as they set the conditions in which the device operates, but are not within the scope of testing as configuration options and combinations are nearly endless. The last items are those outside of control of the user or the manufacturer. The full DFD can be found in fig. 4.1.

Manufacturer items

The items in this group are those whose functionality is provided by the manufacturer. While some configuration options may be left to the user, the bulk of security responsibilities lie within the software and hardware provided. Therefore, the items in this group are the main focus of the testing. The items belonging to this group are shown in red.

The first element of this group would be the device itself (Multiprocess 1.0) as it is responsible for the core functionality, which is the operation of the smart plug functionality. Within it, three interfaces can be extracted from the configuration, which are the discovery service (Process 1.1), the API (Process 1.2) to operate the device and cloud interaction (Process 1.3). Additionally, the key item in the device is the smart plug functionality (Output 1.4). One last element is the firmware of the device (File 1.5) and a process for OTA flashing (Process 1.6). However, no tests targeting the firmware were performed due to the possibility of rendering the device inoperable. Additional detail can be found in fig. 4.2.

The second element of this group, the smartphone and the app (Multiprocess 2.0), is responsible for controlling user input and output. One defining feature of the smartphone is the coexistence of a WLAN interface and a mobile Internet connection, as well as the interaction with different apps. The items are bundled together due to the multitude of environments possible (iOS, several Android versions, etc.) and lack of capabilities for mobile testing, which limited the resolution at which the DFD could be developed.

Lastly, the third element in this group is the manufacturer cloud infrastructure (Input/Output 3.0). The device communicates with the cloud for logging purposes and with the smartphone app for a variety of functionalities such as account authentication and consulting logging information. The cloud could be considered a multiprocess with different elements within like logging files or user information, but has been left out of the scope due to ethical and legal concerns. Nevertheless, security services provided like repudiation are under consideration through the interaction with local elements.

User items

In order to provide its service, the device needs to be provisioned with different elements. Said elements are not within the control of the manufacturer but are critical to the operation of the device. In order to test them accordingly, they have been provided with default configurations and the scope of the testing is the interaction with the device.

One item in this category is the WiFi router (Process 4.0). The device defines a boundary between the WLAN set by the user and the Internet, increasing its exposure. Additionally, it holds a special place in the network as it creates the encrypted channels and handles all traffic.

Another element in this category is the smartphone. Although it is grouped with the application, which belongs to the previous group, the security configuration is within the user's control.

Lastly, some items in this category are not necessary for the device to work, but may exist in the network. While some may be configured to interact with the device (Input 5.0 and the corresponding cloud service in Input/Output 6.0), as is the case for things like voice assistants, others are completely external but have the capability of interacting with the network (Input/Output 7.0).

External elements

The last group is conformed by elements that are entirely out of the control of neither the user nor the manufacturer. Some of these items are necessary for the operation of the device while some are a consequence of the nature of the Internet. These elements are shown in white

The first of the items in this group is the DNS service (Input/Output 8.0), which the device needs in order to resolve cloud domains. While there is a history of attacks to and from DNS servers, connection to them is necessary and risk mitigations do exist.

The second item in this group are Internet elements (Input/Output 9.0)that can interact with any other element or vice versa. This is the initial position of an attacker.

Threats

To conclude the DFD, the STRIDE taxonomy is used to determine the threats for every element. A summary of this section can be consulted on tab. 4.5.

Regarding inputs and outputs, the main concern is item 1.4, or the actuator inside the device, which would be the goal of the attacker under this scenario. Next, interacting elements such as 3.0, 5.0, 6.0 and 7.0 can be the source of spoofing by being able to perform unwanted activities and in a lesser matter the source of repudiation if said activities cannot be attributed. Lastly, the DNS (8.0) service can be spoofed and tampered (if the queries are answered maliciously) and can leak information in some cases. Additionally, DNS attacks can be used to render other services unavailable.

Regarding processes, most are vulnerable to all of the categories in the STRIDE taxonomy, with the exception of the mDNS discovery service (1.1) whose exploitation could lead to information disclosure and render the service unavailable if compromised.

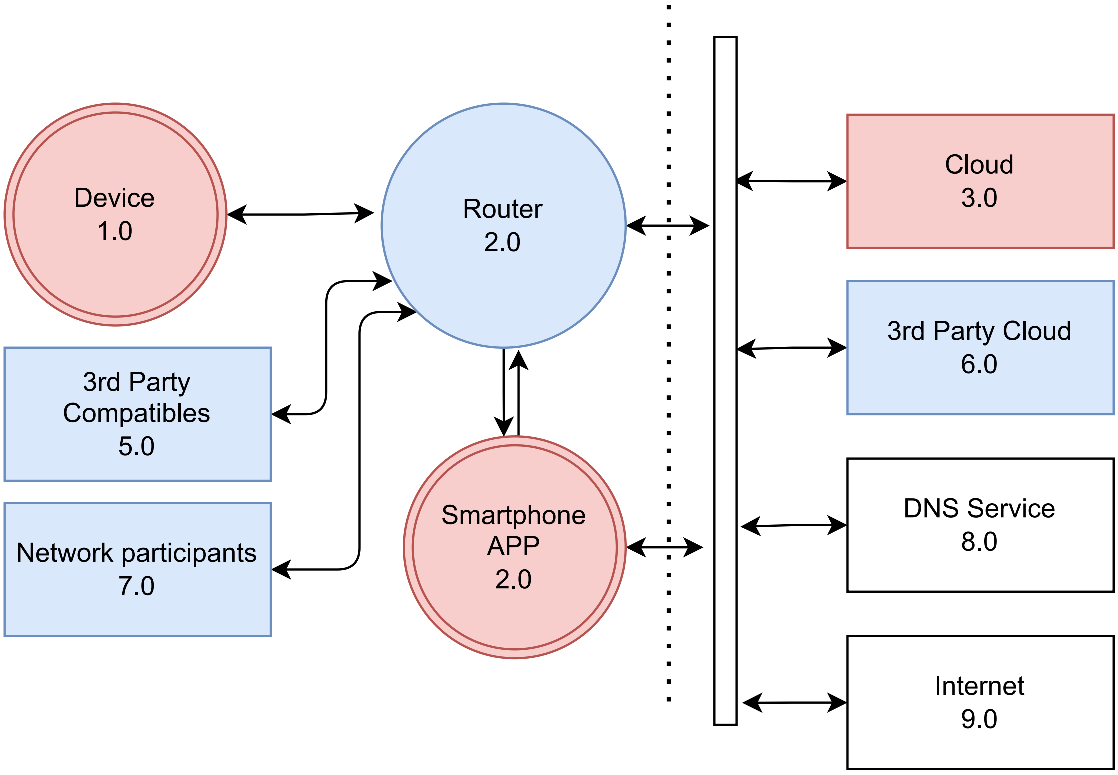


Figure 4.1: Sonoff Mini R2 DFD

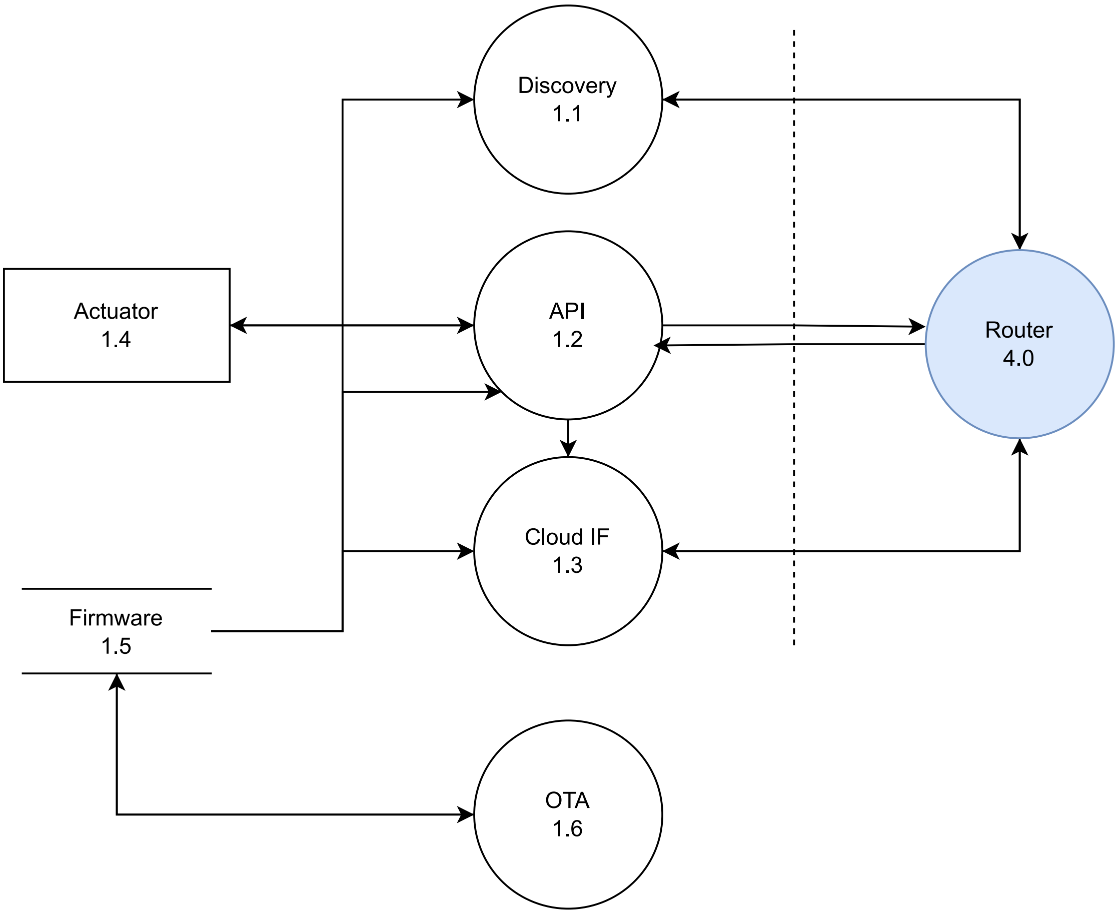


Figure 4.2: Device detail

The firmware file can be tampered with, contain sensitive information or be made unavailable.

Regarding data flows, there are three main groups. The first one, comprised of flows within the device, are essentially unreachable to the attacker. Next, WLAN and Internet flows to authorised users are vulnerable to tampering (channel integrity not granted), information disclosure and denial of service. Lastly, unauthorized elements in both networks cannot be vulnerable since they are not participants.

4.1.3 Controls

As stated previously, the purpose of testing is verifying whether the risk mitigations necessary for an attack are in place. In order to achieve this, an argumentation for the mitigations required is provided in this section. The proposed mitigations were derived from the threat model provided.

The goal in this attack scenario is to change states in the actuator component. In order to achieve this, an attacker would need to induce unwanted behavior in the control API or the cloud interface. Normally, the latter is outbound and, since testing the cloud is not allowed, this strategy of attack is not within the scope of active testing. However, the API interface can be tested thoroughly. Additionally, the discovery service provides essential functionalities that can help in an attack if done incorrectly.

Regarding the cloud interface, it is important to verify that outbound connections use properly secured channels. In the case of the cloud connection, an HTTPS should be employed with proper certificate handling. This protects confidentiality, integrity and authentication on many elements. However, denial of service on this side is arguably unavoidable. Logging should be accessible even if cloud connection is not available, in order to guarantee non-repudiation.

Regarding the API, the traffic should use proper cryptography on all useful parameters, leaving only those that cannot be encrypted in cleartext. Authorization should be protected by preventing replay attacks through robust sequence checks. Lastly, elevation of privilege and denial of service should be avoided by verifying input data against unexpected parameters that could induce unwanted behavior.

Additionally, information broadcast by the discovery service should be limited to the strictly necessary, and any protected information should be encrypted.

Lastly, the smartphone app should conform to standards set by organizations like OWASP.

A summary of all elements with the associated threats can be found on tab. 4.5.

Table 4.5: Sonoff threats table

Element type	Name	Number	Threats
Input/Output	Actuator	1.4	GOAL
	Cloud Service	3.0	S,R
	Third Party Compatibles	5.0	S,R
	Third Party cloud	6.0	S,R
	Network Participants	7.0	S,R
	DNS	8.0	S,T,I,D
Processes	Discovery	1.1	I,D
	API	1.2	S,T,R,I,D,E
	Cloud IF	1.3	S,T,R,I,D,E
	OTA	1.6	S,T,R,I,D,E
	Smartphone and app	2.0	S,T,R,I,D,E
	Router	4.0	S,T,R,I,D,E
Data Stores	Firmware	1.5	T,I,D
Data Flows	Control	(1.2↔1.4)	-
	Execution	(1.5 → 1.1)	-
		(1.5 → 1.2)	-
		(1.5 → 1.3)	-
	OTA	(1.5↔1.6)	-
	Report	(1.2↔1.3)	-
	WLAN	(1.1 ↔ 4.0)	T,I,D
		(1.2 ↔ 4.0)	T,I,D
		(1.3 ↔ 4.0)	T,I,D
		(2.0 ↔ 4.0)	T,I,D
		(5.0 ↔ 4.0)	T,I,D
		(7.0 ↔ 4.0)	N/A
	Internet	(4.0;2.0↔3.0)	T,I,D
		(4.0;2.0 ↔ 6.0)	T,I,D
		(4.0;2.0 ↔ 8.0)	T,I,D
		(4.0;2.0 ↔ 9.0)	N/A

4.2 Testing

For classifying the tests, the following structure has been applied: (i) First, the environments are separated in terms of the setup being employed and labeled with the letter A and a number (e.g. A1);(ii) then, the tests performed in that environment are labeled with a three digit number separated by a hyphen from the environment employed (e.g A1-001); Lastly, pieces of evidence are added with the letter 'E' plus a number (e.g. A1-001-E1). Environments, tests and findings will be provided in this section but evidence is left for an annex due to its volume. A summary of the tests performed can be found on tab. 4.6, consisting of a brief description of each test, along with its findings in relation to the threat model.

Table 4.6: Testing results table

ID	Description	Results	Element	STRIDE
A1-001	Quick Install Mode	Pairing process analyzed	1.1	I
A1-002	Traffic capture and analysis	Device transmits API key	1.2	I
A1-003	Advanced functions testing	Interaction with the cloud	6.0	D
		Cloud connection uses TLS	6.0	OK
A1-004	Packet Replay	API accepts packet	1.2	S
		Action logged as valid	1.2, 6.0	S
A1-005	Replay Packet	Device doesn't check sequences	1.0	T
		The action is logged as legitimate	1.0	R
A1-006	Testing DIY mode	DIY mode is essentially insecure	-	
A2-005	Wireshark mDNS search	Device identifyable through MDNS	1.1	I
A2-006	Parameter fuzzing	"sequence" parameter can be altered	1.2	T
		The logging depends on the "selfApikey" parameter	1.2	T
		Altering encryption turns off the device	1.2	D, S
		Providing large responses may render the device unresponsive	1.0	D
A1-007	DNS Spoofing	TLS use is correct	1.3	OK
		The device can operate without cloud	6.0	D
		The device can operate without logging	1.3	R
A3-008	MobSF Static Analysis	Excessive Permissions	4.0	I, E
		Possible vulnerable certificate	4.0	T,I,D,E
		Possible information leakage	4.0	I,E
		Hardcoded information	4.0	I

4.2.1 Environments

A1: Privileged Network Access

In this scenario an attacker controls the network router and is able to sniff all traffic coming through it. Both the user's phone and the smart device are connected to the compromised access point. This is shown in Fig. 4.3. The main purpose of this environment is to extract relevant information about the device and check controls on the internal side, though it is important to consider that failed checks in this category are harder to exploit from the attack scenario perspective, as an attacker would have to compromise the users' router. The resources employed in this environment were as follow: A normal consumer grade WiFi router, a laptop running Windows 11, a WiFi Pineapple Router, a *Xiaomi Mi Phone 5* Smartphone and the *Sonoff Mini R2*. The purpose of the WiFi router is to isolate the

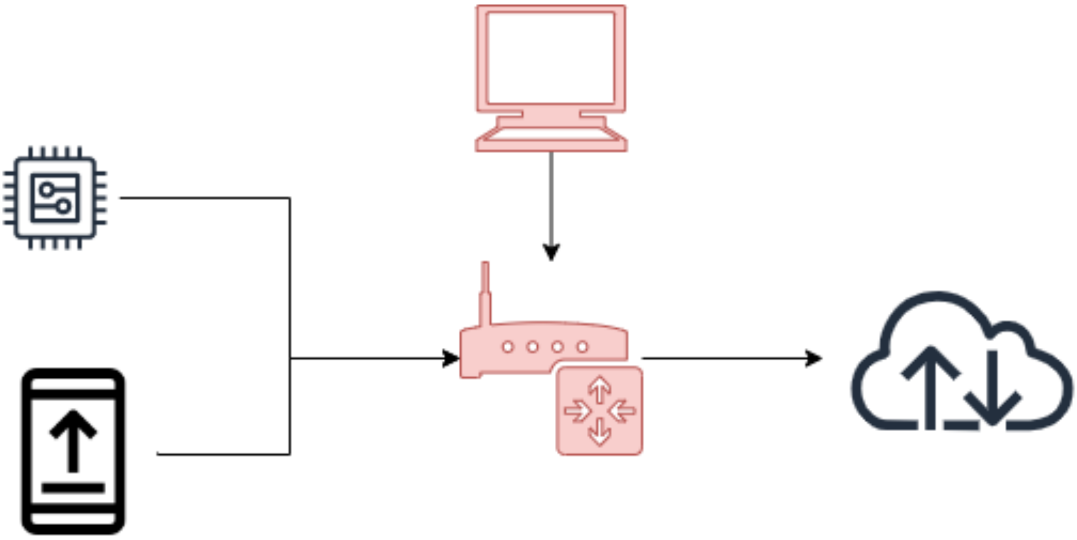


Figure 4.3: Environment A1

network and becoming the only gateway available to the Internet. The computer is then connected to the aforementioned router and relays the connection to the Pineapple. The Pineapple in turn established a monitored network to which both the smartphone and the device connect. Through this configuration, legitimate messages can be sent from the phone to the device and captured by the Pineapple. Then, the analyst can download traffic captures, analyze it with wireshark or send any request to the network. This is summarized in Table 4.7.

Name	Information	Controller
Smart relay	Sonoff Mini R2	Target
Smartphone	Xiaomi Mi Phone 5	User
Pineapple	WiFi Pineapple Mk VII	Attacker
Computer	Windows 11	Attacker
WiFi Router	Linksys E5400	Boundary

Table 4.7: Hardware resources for A1

In addition, the software resources were as follow. On the Pineapple, *tcpdump* was employed to capture traffic and filter it. On the computer, *Wireshark* was used to analyze traffic captures and *Postman* was used to send custom requests to the target device. Lastly, the firmware or operating system is included for the relevant devices. This is shown in Table 4.8.

Name	Location	Version
Hak5 Firmware	Pineapple	2.1.3
tcpdump	Pineapple	4.9.3
libcap	Pineapple	1.9.1
Windows	Laptop	11.0.0
Wireshark	Laptop	4.0.5
Postman	Laptop	10.15
eWeLink	Smartphone	4.32.0
Android	Smartphone	6.0
Sonoff Firmware	Sonoff Mini R2	3.6.0

Table 4.8: Software resources for A1

A2: Non-privileged Network-Access

In this scenario an attacker does not control the network router. The user's phone, the smart device and the attackers machine are connected to the access point. This is shown in Fig. 4.4. The main purpose of this environment is to provide testing from the position of a compromised device in the network, where the attacker may or may not have root access.

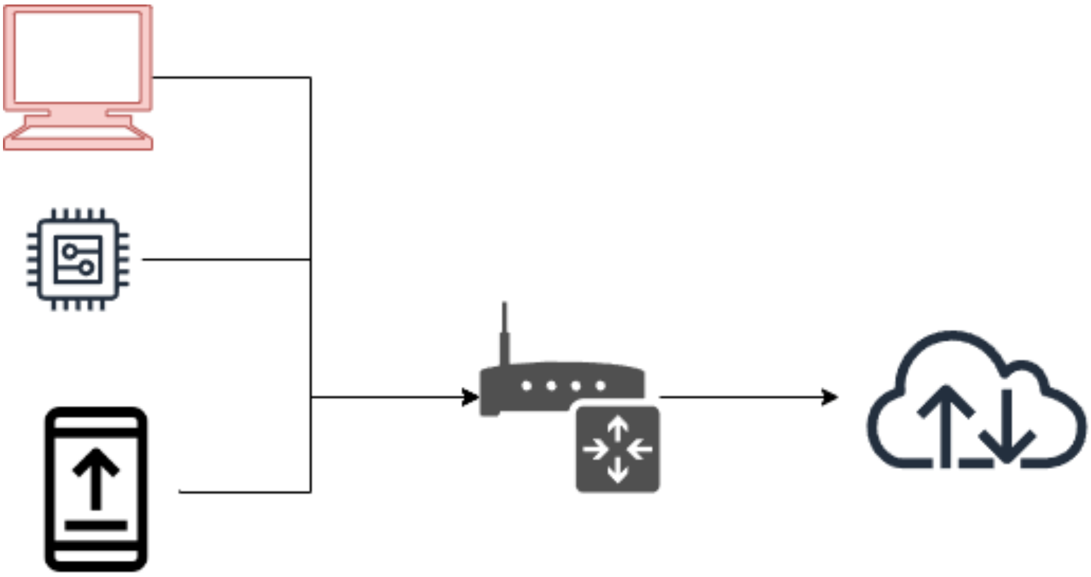


Figure 4.4: Environment A2

The essential difference from scenario A1 lies in the network configuration: the Pineapple is removed and all elements are connected to the basic router.

A3: Software tests

This scenario is entirely virtual. In it, digital resources are employed to analyze or extract information from digital resources, such as software provided by the manufacturer.

4.2.2 Tests

A1-001: Quick Install testing

The purpose of this test process was deploying a working device in the environment and using its default working configuration to verify the information provided in the manual and extract valuable knowledge for future tests. In order to achieve this, the smartphone was connected to the Pineapple network. Then, the app was put in pairing mode by providing the network's SSID and password. Simultaneously, the device was put in search mode and the instructions were followed until the pairing was complete. Lastly, the connection was verified and the information available on the app was analyzed. This is shown in evidence A1-001-E1 (fig. B.1), a screenshot taken on the Pineapple's dashboard showing connected clients. Additionally, a screenshot on the smartphone, shown in evidence A1-001-E2 (fig. B.2), was taken to show the device's running firmware.

Beforehand, *tcpdump* was running on the Pineapple Router filtering for the device on IP 172.16.42.200 by using the filter `tcpdump  -i br-lan host 172.16.42.200`. The output was saved to evidence A1-001-E3 and analyzed.

- The device´s firmware version is 3.6.0 (Evidence: A1-001-E2 found in fig. B.2).
- There is logging data information.
- The device answers to local mDNS queries that include its IP address and DeviceID (Evidence: A1-001-E3). Said mDNS reply contains:
  - Device ID.
  - Device IP.
  - The type of device (plug).
  - Sequence parameter.
  - Encrypted data.

Overall, the test was useful for verifying valuable information about the device. Additionally, the test shows some **information disclosure** on the discovery service (1.1).

A1-002: Traffic capture and analysis

The purpose of this test is to capture traffic using the privileged router position in order to analyze it and extract relevant information about the device's network interaction. In order to achieve this, the Pineapple used *tcpdump* for traffic filtering and formatted output into a file. Then, the file is analyzed using *Wireshark*. The findings were as follow:

- The switching data is sent using JSON through an http channel (Evidence:A1-002-E4). The JSON requests contains the following information:
  - A sequence parameter.
  - Matching device ID.
  - API key.
  - Encrypted data.
- The device performs a TLS connection with an IP that belongs on a block assigned to Amazon. Most likely the cloud infrastructure for the app (Evidence:A1-002-E4).

The test shows correct usage of encryption to communicate commands and information with the server. The device shows, however, **information disclosure** on select parameters by sending them through HTTP, like the "selfApikey" which is personal user information.

A1-003: Test scheduling and other functions

Complex functions were tested to observe the behavior of the device. In order to achieve this, functions outside switching were tested and the information captured similarly to the previous tests. The information gathered was as follow:

- Advanced functionality is handled through the cloud (Evidence:A1-003-E5).
- Scheduling available.

The test shows that advanced functionality is handled through the cloud infrastructure. TLS is used properly but this could entail **denial of service** on the cloud connection (6.0).

A1-004: Replay control packet

The purpose of this test is to assess whether the device is vulnerable to replay attacks by accepting valid packets captured by the attacker and resent from a different device at a different time. In order to achieve this, the JSON data captured in previous tests (Evidence:A1-002-E4) is sent using Postman. The findings were as follow:

- The device switches on after JSON data is resent (Evidence: A1-004-E6 found in fig. B.3).
- The device logs the activity as legitimate (Evidence: A1-004-E7 found in fig. B.4).

This shows that the API has vulnerable **authorization** methods, by allowing a valid packet to be accepted at a different time from its creation by a different person. Additionally, it shows weak **non-repudiation** by transmitting the users API key over unencrypted channels. This information is made available to the network router (2.0) and requires little effort.

A2-005: mDNS wireshark capture

The purpose of this test is assessing whether a non privileged device is able to obtain valuable information about the device from reading mDNS queries on the network. In order to achieve this, wireshark is put on listening mode and set to filter for the protocol. If messages are received, the content is analyzed for the relevant information. The query-response interaction was received and analyzed, obtaining the following results:

- The mDNS queries were captured without interaction.

This findings show a low level **information disclosure** to other devices in the network (7.0) and the network router (2.0), available with little interaction. It can be argued, however, that this functionality is a feature rather than a vulnerability, since the information exchanged is required for the operation.

A2-006: Parameter Fuzz

The purpose of this test is assessing the device's reaction to altered requests. In order to achieve this, the valid message captured in test A1-004 was sent through postman after alterations were made. Then, the reaction of the device was logged and analyzed. Sending the previously captured data turns the device on. However, other types of interactions yield different results, such as error codes or no response.

Parameter	Changes	Result
sequence	Altering one number	Device turns on
	Removing parameter	No response
	Empty parameter	Device turns on
	Sending letters	Device turns on
	29 Numbers	Device turns on but no response
	90 numbers	Device turns on and becomes temporarily unavailable
deviceid	Altering one char	Device turns on
	Removing parameter	No response
	Empty parameter	Device turns on
	Non hex value	Device turns on
	500 numbers	No response
selfApikey	Altering one char	Device turns on with "deleted user" on log
	Removing parameter	No response
	Empty parameter	Device turns on with "APP" on log
	Non hex value	Device turns on with "deleted user" on log
	70 numbers	Device turns on and is briefly unavailable is briefly unavailable
iv*	Change one number	Error 400
	Removing parameter	No response
	Change number to letter	Error 400
		Some letters turn on the device
	Random input	Device is turned off and logged as "device configuration"
encrypt	false	Error 400
	Removing parameter	No response
	Other modifications	Error 400
data*	Removing parameter	No response
	modifying data	No response Error 400

Table 4.9: Fuzzing results

The results are listed in table 4.9. Tests marked with an asterisk were decoded, altered and encoded before sending.

The findings show the device lacks verification for some parameters, which allows **spoofing** from withing the network (7.0) and the network router (2.0). Additionally, the device is vulnerable to manipulation through **denial of service** from the same elements.

A1-007: DNS Spoofing

The purpose of this test was assessing the device's connection with the internet by intercepting DNS requests and returning an alternate IP to act as a proxy between the device and the cloud. In order to achieve this, the elevated privilege router is configured using the dnsspoof module to route any DNS request to a given domain name to a Burp Suite proxy configured to send it to its valid destination.

After the domain has been extracted, the router can be configured to answer to DNS queries for that domain with the IP of the analyst's machine. Simultaneously, Burp is configured to listen to port 443 on the machine and to forward all requests to the original domain IP. After the configuration was finished, the pairing process is initiated with traffic capture enabled. Next, the behavior and captures were reviewed to analyze the device's mitigations for this attack.

The traffic analysis shows that the connection to the proxy is successfully refused by the device, indicating a proper handling of security in regards to protecting the channel with the cloud, but continues the process indicating that the device continues to provide functionality without the necessary resources to provide some security features.

The findings show that the device can provide proper authentication, integrity and confidentiality on the connection with the cloud. However, the analysis shows a compromise on **availability** on said channel which in turn causes a compromise on **non-repudiation**.

A3-008: MobSF Static Analysis

In order to test the application security, a static analysis of the compressed code has been performed using the tool MobSF. The analysis gave the app a high risk rating, including 8 high and 13 medium findings (File summary in tab. B.1). However, automated analysis tools are known to be prone to false positives, so careful review of the results was performed. The relevant findings are as follow:

- Some permissions are marked as dangerous. Access to geolocation is requested and arguably not necessary and risky, as it could be used to fine-tune an attack to target a specific area. Additionally, access to external storage and other information on the phone, which might be used by an attacker to elevate privileges. Lastly, the analysis marks camera or audio access as risky. However, those can be considered as necessary for some features.

- The certificate is marked as vulnerable to Janus vulnerability on certain Android versions. This risk is low as a significant amount of knowledge and resources would be needed to exploit this in a reduced, and arguably shrinking, number of targets.
- Some features of the manifest have been marked but are arguably a necessary for the functioning of the application like HTTP traffic or broadcast receiving. Nevertheless, other features have been marked due to possible exfiltration of information to other elements in the phone. However, this needs to be analyzed in full detail to ascertain their exploitability.
- The code analysis suggests different weaknesses like hardcoded credentials, weak cryptographic algorithms, insecure security features implementation and raw SQL queries that could potentially involve an injection attack. However, some of these indications may not be exploitable, requiring dynamic testing to assert their reach.
- The analysis shows hardcoded information such as a *Google* API key and a *Firebase* database URL.

All findings point to potential failures that could be employed in a MaDIoT attack. However, this test provides no material evidence without further analysis of the implementation in context.

4.3 Results

The summary of the results can be found on table 4.6. Overall, the device shows points of failure that could potentially be exploited in a MaDIoT attack. The following proof of concept has been developed in order to prove the exploitability of the device.

Attacking the device through a vulnerable outward facing application

For producing a proof of concept of the exploitability of the machine, the following testing scenario has been setup.

The first step step was setting up the environment. In order to achieve this, the device was installed using the regular smartphone and router from scenario A1 to create a working environment. Additionally, the computer in the network was connected to the network and configured to launch the following python exploit found on lst. 4.1.

Listing 4.1: Python exploit

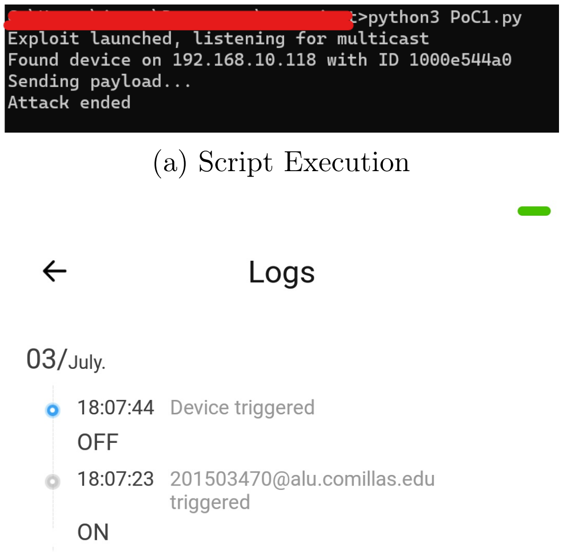
```
from scapy.all import sniff, IP, UDP, DNS
import requests
import json

def escuchar_dispositivo(timeout=5):
    def packet_callback(packet):
        if IP in packet and UDP in packet and DNS in packet:
            ip_src = packet[IP].src
            dns = packet[DNS]
            if dns.qr == 1 and dns.opcode == 0 and "ewelink" in str(dns):
                matching_ips.add(ip_src)
                domain_name = dns.an.rdata.decode("utf-8")
                if "eWeLink_" in domain_name:
                    device_number = domain_name.split("eWeLink_")[1].split("._")[0]
                    device_numbers.add(device_number)

    matching_ips = set()
    device_numbers = set()

    sniffed_packets = sniff(filter="udp port 5353", prn=packet_callback, timeout=timeout)

    if device_numbers:
        return matching_ips.pop(), device_numbers.pop()

    return None, None

def exploit(ip, device_id):
    url = f"http://{ip}:8081/zeroconf/switch"

    data = {
        "sequence": "1234567891011",
        "deviceid": f'"{device_id}"',
        "selfApikey": "cafebabe-6633-0307-2023-deadbeef",
        "iv": r"MzY2NjMwNDE2NjQxNTA3NACUM==",
        "encrypt": True,
        "data": r"9x2PEuxgwKFYqK/UHb9+EA=="
    }

    try:
        print("Sending payload...")
        requests.post(url, json=data, timeout=None)
    except requests.exceptions.RequestException as e:
        print("Attack ended")
        exit(0)


print("Exploit launched, listening for multicast")
ip, device_id = escuchar_dispositivo()

if ip:
    print(f"Found device on {ip} with ID {device_id}")
    exploit(ip, device_id)
```

This exploit works by listening to an mDNS reply that contains the keyword for *eWelink*. Then, the packet is scraped to find the original IP and the device's ID. Lastly, a JSON POST request with spoofed parameters is sent to shut the device down. Additionally, it can be seen that the device is unable to attribute the action to anyone but itself. This operation is shown in fig. 4.5. This proof of concept is built from findings extracted from tests *A2-007* and *A2-008*.

4.3.1 Proposed solutions

For the device, the most reasonable measures for solving the flaws found could be: (i) Verifying message integrity and sequence inside the encrypted parameter and (2) employ memory safe functions to prevent overflow attacks that could affect the device.

Figure 4.5: Automated exploitation



```
[redacted]>python3 PoC1.py
Exploit launched, listening for multicast
Found device on 192.168.10.118 with ID 1000e544a0
Sending payload...
Attack ended
```

(a) Script Execution

Logs

03/July.

18:07:44 Device triggered
OFF

18:07:23 201503470@alu.comillas.edu triggered
ON

(b) Logging

Chapter 5

Conclusions and Future Work

This chapter provides a description of the relevant knowledge and experience that was a result of the development of the project along with a proposition of future lines of work that could be applied to the project in future steps or on similar projects.

5.1 Conclusions

First and foremost, the project has proven that the proposed methodology for threat modelling is relevant to the body of research in cybersecurity, as proven by the fact of its inclusion during JNIC 2023. Additionally, the present document proves that the proposed methodology is useful for guiding testing for cybersecurity in devices. Lastly, the work performed shows that devices employed to control HVAC systems, like smart plugs, lack cybersecurity measures, as evidenced by the device analyzed. This is merely a stepping stone on the path to securing EPES from the consumption field but proves reasons for concern.

As a final note, work has been the application of several skills acquired during the course of a Master´s Degree. First, knowledge on thread modelling has been applied to develop the methodology, in addition to practical testing skills to bring the assessment forward. Additionally, knowledge of networking and computation have been necessary to fulfill the technical requirements for the project.

5.2 Future Work

The work performed shows a few defects that could be addressed in future work, namely: (i) the methodology's last steps take big leaps when it comes to control definition and higher support on established on reputed specifications for controls should be helpful, (ii) There is a lack of visibility of the device's operation when performing the modelling and a boilerplate set of tests (such as installation outcome analysis) on the generic steps of the methodology could help with a parallel strategy for starting tests and finishing the threat model and lastly (iii) the structure of a DFD is not reflective of security with a specific attack in scope. Additionally, the time and resources for testing firmware and application were limited.

5.2.1 Firmware emulation

One possible route to advance the line proposed in this project could be performing testing on emulated firmware. This would have the advantage of being able to test products before purchase, potentially making the assessment of different device less expensive. Additionally, it could potentially help automate testing, which could be helpful in keeping up with the fast development of the market. Nevertheless, this emulation is no easy achievement.

5.2.2 Application in-depth analysis

A different way to continue with the assessment of this device and gain insights and experience for others would be performing deeper analysis on the application. This could have the advantage of having a more complete analysis on one of the most exposed elements of the model.

Additionally, analyzing different applications would be an effective way to assess the risk of Smart HVAC Systems, since the number of applications available is smaller than the number of devices in the market.

Bibliography

[1] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[2] T. S. S. of ITU, "Recommendation itu-t y. 2060: Overview of the internet of things," *Series Y: Global information infrastructure, internet protocol aspects and next-generation networks-Frameworks and functional architecture models*, 2012.

[3] European Parliament and The Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance)," 2016, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679.

[4] J. Fuster, S. Solera-Cotanilla, J. Pérez, M. Vega-Barbas, R. Palacios, M. Alvarez-Campana, and G. Lopez, "Analysis of security and privacy issues in wearables for minors," *Wireless Networks*, pp. 1–17, 2023.

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th {USENIX} security symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.

[6] Jefatura del Estado, "Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas," 2011, https://www.boe.es/buscar/pdf/2011/BOE-A-2011-7630-consolidado.pdf.

[7] BBC, "Ukraine power cut 'was cyber-attack'," *BBC News*, 2017. [Online]. Available: https://www.bbc.com/news/technology-38573074

[8] eFORT, "Newsletter," 2023. [Online]. Available: https://2275720.hs-sites.com/efort-newsletter-1

[9] S. Soltan, P. Mittal, and H. V. Poor, "BlackIoT: IoT botnet of high wattage devices can disrupt the power grid," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 15–32. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/soltan

[10] B. Huang, A. A. Cardenas, and R. Baldick, "Not everything is dark and gloomy: Power grid protections against iot demand attacks," in *usenix security symposium*, 2019, pp. 1115–1132.

[11] T. Shekari, A. A. Cardenas, and R. Beyah, "{MaDIoT} 2.0: Modern {High-Wattage}{IoT} botnet attacks and defenses," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 3539–3556.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[13] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 449–453.

[14] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[15] S. Banerji and R. S. Chowdhury, "On IEEE 802.11: Wireless lan technology," *International Journal of Mobile Network Communications and Telematics*, vol. 3, no. 4, pp. 45–64, aug 2013. [Online]. Available: https://arxiv.org/abs/1307.2661

[16] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 180–189.

[17] M. Caneill and J.-L. Gilis, "Attacks against the wifi protocols wep and wpa," *Journal, no. December*, 2010.

[18] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*. ACM, 2017.

[19] C. P. Kohlios and T. Hayajneh, "A comprehensive attack flow model and security analysis for wi-fi and wpa3," *Electronics*, vol. 7, no. 11, 2018. [Online]. Available: https://www.mdpi.com/2079-9292/7/11/284

[20] Cyber resilience act. European Comission. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act

[21] A. Dabrowski, J. Ullrich, and E. R. Weippl, "Grid shock: Coordinated load-changing attacks on power grids: The non-smart power grid is vulnerable to cyber attacks as well," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 303–314.

[22] MalwareMustDie, "Mmd-0056-2016 - linux/mirai, how an old elf malcode is recycled.." 2016. [Online]. Available: https://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html

[23] OWASP, "OWASP Internet of Things," https://owasp.org/www-project-internet-of-things/, accessed: May 19, 2023.

[24] J. Serra, D. Pubill, A. Antonopoulos, and C. Verikoukis, "Smart hvac control in iot: Energy consumption minimization with user comfort constraints," *The Scientific World Journal*, vol. 2014, 2014.

[25] A. Lautenbach, M. Almgren, and T. Olovsson, "Proposing heavens 2.0–an automotive risk assessment model," in *Proceedings of the 5th ACM Computer Science in Cars Symposium*, 2021, pp. 1–12.

[26] M. Howard and S. Lipner, *The security development lifecycle.* Microsoft Press Redmond, 2006, vol. 8.

[27] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover security design flaws using the stride approach," Nov 2006. [Online]. Available: https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach

[28] "Energy consumption in households," June 2022. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households#Energy_consumption_in_households_by_type_of_end-use

[29] Sonoff, "Sonoff Mini R2 Product Specification," https://sonoff.tech/wp-content/uploads/2021/03/%E4%BA%A7%E5%93%81%E5%8F%82%E6%95%B0%E8%A1%A8-MINIR2-20201118.pdf, accessed: May 19, 2023.

[30] ——, “Sonoff Mini R2 User Manual,” https://sonoff.tech/wp-content/uploads/2021/03/%E4%BA%A7%E5%93%81%E5%8F%82%E6%95%B0%E8%A1%A8-MINIR2-20201118.pdf, accessed: May 19, 2023.

[31] ——, “Sonoff Mini R2 Quick Start,” https://sonoff.tech/wp-content/uploads/2021/03/%E4%BA%A7%E5%93%81%E5%8F%82%E6%95%B0%E8%A1%A8-MINIR2-20201118.pdf, accessed: May 19, 2023.

[32] ——, “DIY Developer,” https://sonoff.tech/diy-developer/, accessed: May 19, 2023.

[33] ITEAD, “Sonoff Devices DIY Tools,” https://github.com/itead/Sonoff_Devices_DIY_Tools/blob/master/SONOFF%20DIY%20MODE%20Protocol%20Doc%20v2.0%20Doc.pdf, accessed: May 19, 2023.

[34] U. Nations, “Sustainable development goals,” https://sdgs.un.org/goals, 2023.

Appendix A

Alignment with the SDGs

This project is aligned with the United Nation's Sustainable development goals. Its primary focus are the objectives nine, eleven and seventeen. Additionally, goals seven and twelve are tangentially covered.

The mentioned goals [34] are:

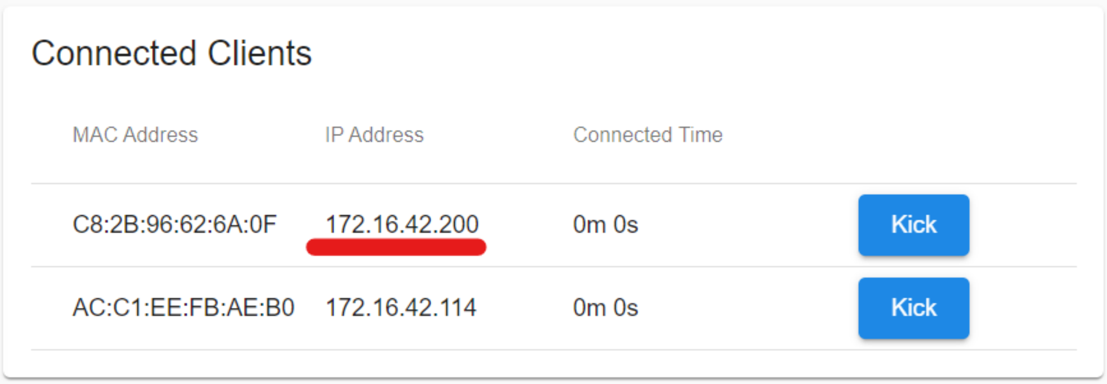
- Goal 7: Affordable and clean energy.
- Goal 9: Industry, innovation and infrastructure.
- Goal 11: Sustainable cities and communities.
- Goal 12: Responsible consumption and production.
- Goal 17: Partnership for the goals.

The main argument for the project's alignment with the aforementioned ninth and eleventh goals is its contribution to the security of the infrastructure, and consequentially contributing to their resilience both in the infrastructure itself and the communities for whom it provides. Moreover, this project is aligned with the seventeenth goal as it is a part of a wider project, fruit of the collaboration of different entities.

Furthermore, this project can be ascribed secondarily to goals seven, due to the significance of electrical power in the project, and twelve, as it can contribute to safer developments in new consumer tech.

Appendix B

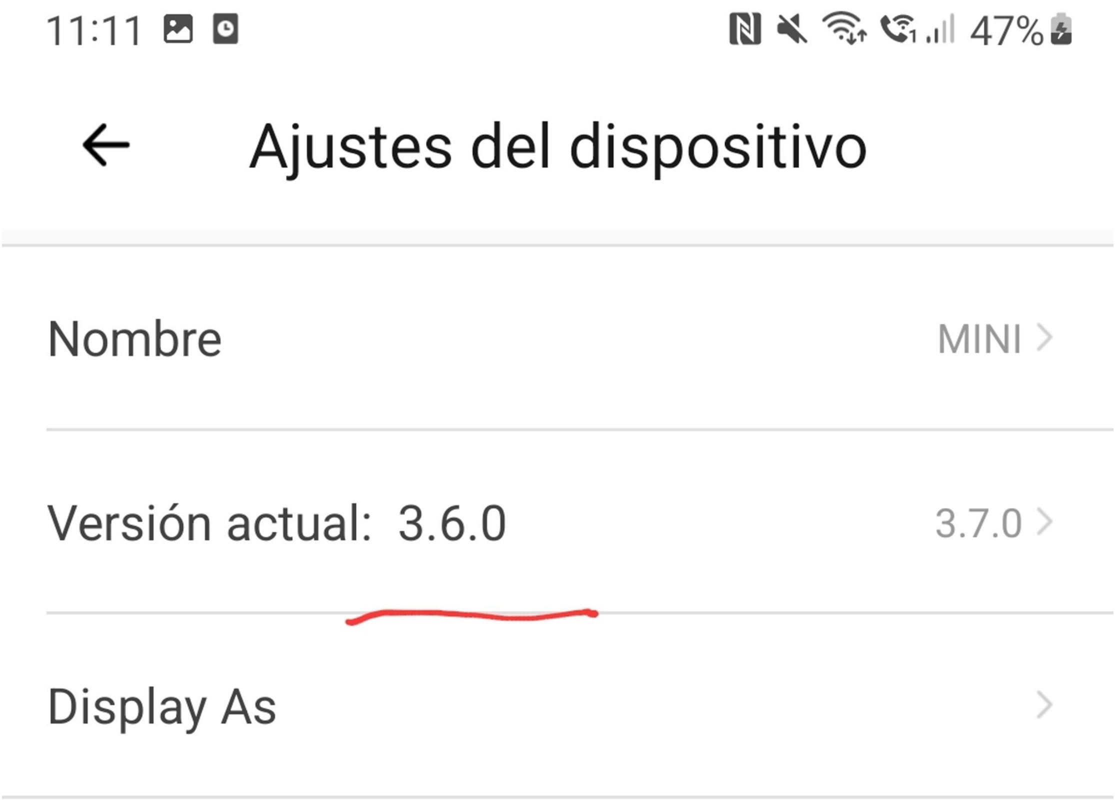
Evidence



Connected Clients

MAC Address	IP Address	Connected Time	
C8:2B:96:62:6A:0F	172.16.42.200	0m 0s	Kick
AC:C1:EE:FB:AE:B0	172.16.42.114	0m 0s	Kick

Figure B.1: Evidence A1-001-E1



11:11 47%

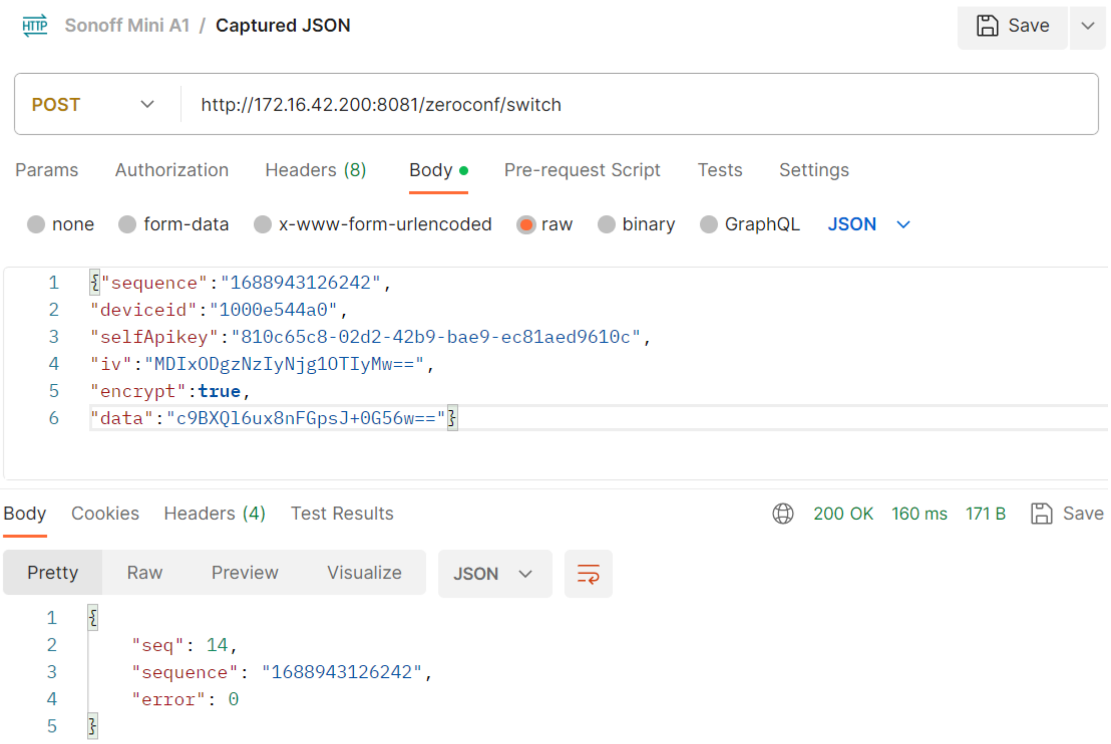
Ajustes del dispositivo

Nombre MINI

Versión actual: 3.6.0 3.7.0

Display As

Figure B.2: Evidence A1-001-E2



POST http://172.16.42.200:8081/zeroconf/switch

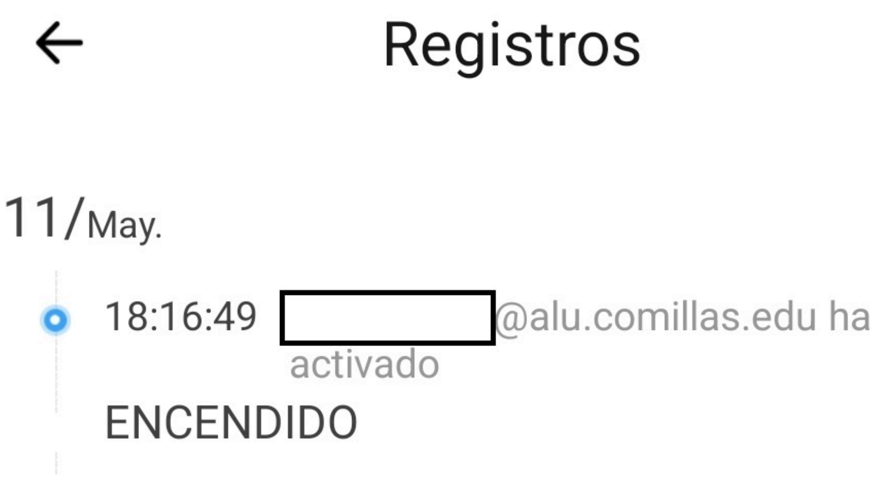
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{"sequence":"1688943126242",
"deviceid":"1000e544a0",
"selfApikey":"810c65c8-02d2-42b9-bae9-ec81aed9610c",
"iv":"MDIxODgzNzIyNjg1OTIyMw==",
"encrypt":true,
"data":"c9BXQl6ux8nFGpsJ+0G56w=="}
```

200 OK 160 ms 171 B

```
{
    "seq": 14,
    "sequence": "1688943126242",
    "error": 0
}
```

Figure B.3: Evidence A1-004-E6



Registros

11/May.

18:16:49 [redacted]@alu.comillas.edu ha activado ENCENDIDO

Figure B.4: Evidence A1-004-E7

Finding	Type	Severity
ACCESS_FINE_LOCATION	Permission	Dangerous
ACCESS_COARSE_LOCATION	Permission	Dangerous
READ_EXTERNAL_STORAGE	Permission	Dangerous
WRITE_EXTERNAL_STORAGE	Permission	Dangerous
GET_TASKS	Permission	Dangerous
READ_LOGS	Permission	Dangerous
MOUNT_UNMOUNT_FILESYSTEMS	Permission	Dangerous
WRITE_SETTINGS	Permission	Dangerous
Application Vulnerable to Janus	Certificate	Warning
App can be installed on vulnerable Android version	Manifest	Warning
Launch Mode of activity (*) is not standard	Manifest	High
Task affinity is set for activity (*)	Manifest	Warning
Launch Mode of activity (*) is not standard	Manifest	High
Files may contain hardcoded sensitive information	Code Analysis	Warning
SHA-1 is a weak hash known to have collisions	Code Analysis	Warning
MD5 is a weak hash known to have collisions	Code Analysis	Warning
The app uses an insecure RNG	Code Analysis	Warning
App can read/write to External Storage	Code Analysis	Warning
IP Address disclosure	Code Analysis	Warning
App uses SQLite database and executes raw SQL query	Code Analysis	Warning
The app uses encryption mode CBC with PKCS5/PKCS7 padding	Code Analysis	High
App creates temp file	Code Analysis	Warning
Insecure WebView implementation	Code Analysis	High
Weak encryption algorithm used	Code Analysis	High
The app uses ECB mode in cryptographic encryption algorithm	Code Analysis	High
Some libraries don't implement Stack canaries	Libary analysis	High
google_api_key	Hardcoded secrets	Undetermined

Table B.1: Evidence A3-008-E9