

# Developing digital evidence gathering techniques for IoT devices

Shiluvelo Makhubele<sup>1,2\*</sup> and Sheunesu Makura<sup>2,3</sup>

<sup>1\*</sup>Department of Computer Science, University of Pretoria, Lynnwood Rd, Pretoria, 0002, Gauteng Province, South Africa.

<sup>2</sup>Department of Computer Science, University of Pretoria, Lynnwood Rd, Pretoria, 0002, Gauteng Province, South Africa.

\*Corresponding author(s). E-mail(s): [u19086352@tuks.co.za](mailto:u19086352@tuks.co.za);  
Contributing authors: [makura.sm@up.ac.za](mailto:makura.sm@up.ac.za);

## Abstract

In the rapidly evolving world of the Internet of Things (IoT), ensuring digital forensic readiness is of paramount importance, particularly given the growing integration of smart devices in daily life. This research study examines the digital forensic readiness of Apple's Home app, a central component of the HomeKit smart home ecosystem, which enables users to control various smart devices. By simulating IoT environments with Homebridge-enabled devices, this research study applies ISO/IEC 27043 digital forensic processes, from evidence identification to data handling and storage. The research includes the development of a prototype which we name "SecureCollect" and any mention of "SecureCollect" will refer specifically to this prototype. SecureCollect is used for pre-incident collection and experimentation to assess data storage, extraction, and analysis capabilities. Through SecureCollect, we evaluate security, data integrity, and the challenges associated with digital forensic readiness within the IoT context. The results highlight gaps in forensic readiness, particularly in the management of non-HomeKit-compatible devices, and propose improvements to forensic practices to address these shortcomings.

**Keywords:** IoT, Digital Forensics readiness, Apple Home App, Homekit, Homebridge, IoT Forensic, Security, Evidence Collection

# 1 Introduction

In an era where the presence of Internet of Things (IoT) has become increasingly prevalent, the need for robust digital forensics readiness has become equally important. The Internet of Things (IoT) is an evolving concept that enables electronic devices and sensors to communicate via the internet, utilizing smart devices and online connectivity to address a range of challenges.[1].This interconnected nature of devices has led to an explosion of data and has necessitated the need for robust mechanisms for managing and securing this information [2].

As IoT devices become more integrated into our everyday lives, security has become a growing concern. These devices usually collect and send sensitive information, which makes them targets for cyber crimes and attacks. Ensuring the security of IoT devices involves implementing measures to protect data integrity, confidentiality, and availability. Security protocols must address potential vulnerabilities, including unauthorized access, data breaches, and malware attacks [3]. The need for security extends to all components of the Internet of Things ecosystem, from individual pieces of hardware to cloud services and transmission networks [4].

Digital forensics entails investigating and analyzing digital devices to discover evidence pertaining to criminal activities, security breaches, or other incidents. In the context of IoT, digital forensics faces unique challenges due to the diversity and complexity of devices, the large amount of data generated, and the need for specialized tools and techniques to extract and analyze this data [5].

Digital forensic readiness pertains to an organization or system's capacity to promptly and proficiently handle digital forensic investigations. This includes establishing procedures, guidelines, and technologies to guarantee swift and accurate collection, preservation, and analysis of digital evidence. In the context of IoT devices, it entails designing devices and platforms with forensic aspects in consideration, such as secure logging, data retention protocols, and adherence to legal and regulatory standards. [6]. Within the IoT world lies the concept of smart home platforms which has become common, especially within the notion of smart home technology. These smart home platforms have been proposed by many companies to help manage the ever-increasing number of Iot smart home devices that are getting installed[7]. Some of these platforms are Apple's HomeKit, Amazon AWS (Amazon Web Services) IoT, Samsung's smartThings, IBM Watson IoT Platform[8] and DIY Platforms [7]. The app's ability to store and process data from various devices makes it a critical point of interest for digital forensic investigations, especially in understanding how data is generated, stored, and transmitted within the ecosystem. This research focuses on the forensic readiness of Apple's Home app, examining how data can be identified, extracted, and utilized for forensic purposes, and addressing the unique challenges posed by IoT environments[9].

Apple HomeKit is Apple's smart home framework, designed to allow users to control a variety of connected devices through a unified interface, typically the Home app. The HomeKit infrastructure is built around several key components that provide

secure, efficient communication between devices and the Apple ecosystem. At the core of HomeKit is the Home app, which acts as a centralized hub, allowing users to control smart devices like lights, locks, cameras, sensors, and thermostats. These devices are all interconnected via a secure, encrypted communication protocol that ensures both privacy and data integrity. HomeKit-enabled devices are required to use Apple's authentication chips and must undergo strict security and privacy checks before being certified for use within the HomeKit ecosystem[10].

Homebridge is an open-source software platform that enables the integration of non-HomeKit-compatible devices into the HomeKit ecosystem. Unlike Apple-certified devices, which come with native HomeKit support, many smart home products do not officially support HomeKit. Homebridge acts as a bridge, making these devices appear as HomeKit-enabled accessories. While HomeKit offers a secure and seamless ecosystem, it is somewhat restrictive due to its reliance on Apple-certified devices. While HomeKit provides a limited scope for forensic investigation (due to Apple's stringent security protocols), Homebridge simulates a HomeKit-like environment, which will allow us to explore how forensic readiness can be achieved even with devices that do not natively support Apple's ecosystem[11].

In this research, we demonstrate the digital forensic readiness process as defined by the ISO27043 standard. This was all be demonstrated on Apple's Home app using homebridge enabled devices. We employ different methodologies including a literature study to help understand existing methodologies and tools used for data extraction and evidence gathering especially in an IoT environment, a prototype(SecureCollect) was developed to aid with the digital forensic readiness process , experiments were also be performed using the SecureCollect , Apple's home app and a physical device that are homebridge complaint.

The remainder of this chapter is structured as follows: first, the problem statement is outlined in subsection 1.2 ,then the research questions in 1.3, 1.4 discusses the research motivation, the aim of the research and limitations that must be dealt with during the research are mentioned in 1.5 and this followed by the research methodology in 1.6 and finally the layout of this research study is discussed in subsubsection 1.7. This is then followed by the Chapter 2, 3, 4, 5 and 6.

As the adoption of IoT technology in smart homes grows, concerns around security, privacy breaches, and legal issues become more pressing. However, there is a significant gap in the knowledge and tools necessary to ensure digital forensic readiness for these environments. The current lack of effective forensic mechanisms hampers the ability to collect, preserve, and analyze digital evidence when investigating security incidents or legal matters. This research addresses these challenges by exploring forensic readiness strategies within Apple's HomeKit ecosystem, identifying effective data extraction methods, and enhancing incident investigation processes.

The research aims to answer the following questions:

- 1 **What are the potential vulnerabilities and security threats specific to Apple's Home app and its integration with IoT devices, and how can these be mitigated through forensic readiness ?**

This question aims to identify and analyze potential security vulnerabilities and threats that affect Apple's Home app and its ecosystem of IoT devices. It will explore how forensic readiness techniques can be employed to assess these vulnerabilities, detect security breaches, and recommend mitigation strategies.

- 2 **How can data be extracted from Apple's Home app and used for digital forensics purposes?**

The question aims to survey the methodologies, techniques, and tools within reach for extracting information from Apple's Home app ecosystem for digital forensic analysis. It delves into the process of accessing and retrieving various types of data stored within the Home app, including device configurations, activity logs, user interactions, and communication protocols. Additionally, it examines the forensic implications of extracted data in investigating security incidents involving IoT devices.

By investigating the extraction methods and potential forensic data/evidence within the Home app ecosystem, the research seeks to provide insights into the forensic capabilities and limitations of Apple's Home app as a source of digital evidence.

This question also aims to explore the various elements comprising Apple's Home app ecosystem, such as the Home app itself, HomeKit framework, compatible smart devices, and associated cloud services. By understanding the architecture and functionalities of these components, the research seeks to evaluate their implications for digital forensic readiness and Apple's home app overall Security.

- 3 **What specific challenges and opportunities arise when conducting digital forensics in the context of Apple's Home app and IoT environments?**

This question delves into the specific challenges and prospects found in performing digital forensic investigations within the Apple Home app ecosystem and broader IoT environments. It seeks to identify and analyze challenges such as data encryption, proprietary protocols, cloud-based storage, and device interoperability, which may complicate forensic analysis.

- 4 **What are the potential vulnerabilities and security threats specific to Apple's Home app and its integration with IoT devices, and how can these be mitigated through forensic analysis?**

This question aims to identify and analyze potential security vulnerabilities and threats that affect Apple's Home app and its ecosystem of IoT devices. It will explore how forensic analysis techniques can be employed to assess these vulnerabilities, detect security breaches, and recommend mitigation strategies.

- 4 **How can the SecureCollect ensure the secure collection and transmission of forensic data from IoT devices connected to Apple's Home app and also how can it be designed to support the forensic investigation of various types of IoT devices and data formats within the Apple Home app ecosystem?**

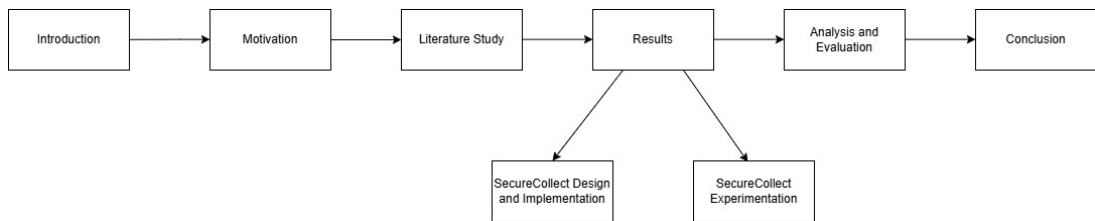
This question focuses on the security measures that need to be implemented in SecureCollect to protect the integrity and confidentiality of forensic data. It will investigate the use of encryption, secure communication protocols, and authentication mechanisms to prevent unauthorized access and data tampering. The last part of this question focuses on the versatility of SecureCollect in supporting diverse IoT devices and data formats. It will explore how to design SecureCollect to accommodate different device types, communication protocols, and data structures to ensure comprehensive forensic analysis.

IoT has increasingly attracted cyber criminals' attention, exemplified by incidents like the 2016 Mirai-related cyber attacks. During these attacks, over 4,000 IoT devices were compromised by the Mirai malware, which was then employed to launch distributed denial of service (DDoS) attacks. These attacks led to disruptions on various well-known platforms [12]. In 2019, the International Data Corporation (IDC) projected that IoT devices would generate approximately 79 zettabytes of data by 2025. This data has the potential to be acquired and utilized in digital forensic investigations. [12].

The research aims at developing a prototype tool for extracting and analyzing data from Apple's Home app to enhance digital forensic readiness in IoT environments. It seeks to provide insights into the type of information that could potentially be extracted from Apple HomeKit app and how this data can be used to investigate security incidents, privacy breaches, and legal matters. The study aims to inform and improve forensic practices by identifying best practices and potential challenges in extracting and analyzing IoT data.

One major limitation of the research is that the prototype may not cover all types of data or all potential sources of evidence within the Home app ecosystem due to technical and resource constraints. The research is specific to Apple's HomeKit environment and may not be directly applicable to other smart home platforms.

The whole research study is layered according to the diagram in figure 1 below



**Fig. 1:** Research Paper Layout

### ***chapters 1: Introduction***

This chapter introduces the research context, problem statement, objectives and layout of the research. It outlines the growing significance of IoT forensic readiness, particularly in smart home environments, and presents research questions that guide the study.

### ***chapter 2: Methodology***

The methodology provides a brief description of how the research was conducted and also justifies why such methods were chosen for the study.

### ***chapter 3: Literature Study***

This chapter provides a comprehensive review of existing research on digital forensic readiness, IoT security, and smart home ecosystems, identifying gaps and framing the study's contribution.

### ***chapter 4: Results***

#### ***SecureCollect Design and Implementation***

This chapter describes the design and development of the forensic readiness prototype, detailing the methodologies used, tools employed, and the specific features tailored for Apple's HomeKit.

#### ***SecureCollect Experimentation***

In this chapter, experimental setups, data collection methods, and testing procedures are outlined. The results from simulations and real-world tests are presented, demonstrating SecureCollect's performance

### ***Chapter 5: Critical Evaluation***

This chapter evaluates SecureCollect's effectiveness in addressing the research questions, critically assessing its strengths and limitations. Future research directions are also proposed based on the findings.

### ***Chapter 6: Conclusion***

This chapter summarizes the key findings from the research, discussing how the research questions were addressed and the significance of the results. It provides a final reflection on the effectiveness of SecureCollect in enhancing IoT forensic readiness and highlights the main contributions of the study. Additionally, this chapter offers suggestions for future research directions, focusing on expanding the forensic capabilities of IoT devices and exploring emerging challenges in smart home environments.

## **2 Methodology**

The research methodology makes use of a mixed-methods approach, integrating qualitative and quantitative techniques to meet the research objectives. This encompasses of the following:

## **2.1 Literature study**

A Literature study is essential for gaining a comprehensive understanding of the existing body of knowledge related to digital forensic readiness in the Iot, Digital forensic readiness and cyber-security at large. It has also help to identify gaps, trends, and key findings from previous studies, which has informed and guided this particular research study.

## **2.2 Prototype**

This provides a practical demonstration of the concepts being studied. It allows for a tangible representation of the theoretical ideas and facilitates the experimentation. This Prototype is referred to as SecureCollect, it is what we use to conduct the digital forensic process on devices connected to Apple's home app to assess the data that is stored in an effort to facilitate future forensic investigations.

## **2.3 Experimentation**

Conducting controlled experiments to collect data under realistic conditions, ensuring integrity and reliability. Normal user interactions are simulated with the Home app and devices, including adding/removing devices, configuring settings, and generating activity logs, network traffic between the Home app and devices is monitored and recorded to identify communication patterns and potential data leakage points. A DoS attack was also simulated to test SecureCollect's incident detection capabilities.

## **2.4 Analysis and evaluation**

Both quantitative and qualitative analysis of the data we got are conducted, and assist in reaching a conclusion. Security gaps of Home app's forensic readiness are identified and improvements are also proposed based on the findings..

## **3 Literature Study**

In this section, a literature study is conducted, this section also aims to explore existing research in the field of digital forensic, computer security and the IoT and consequently identify gaps within the field, this will help guide this research.

### **3.1 Iot and smart apps**

Traditional internet provided wireless connectivity computers around the world but the field has grown so much since its inception in the early 80s, which has led to a new phenomena where every device around us can be made digital and consequently connect to the internet through wireless technology, this phenomena has been coined the internet of things(IoT)[13]. The IoT is a fast evolving field that has significantly impacted different sectors, including automated home applications. Smart home apps, which integrate IoT devices, have revolutionized how users interact with their home environments, offering enhanced convenience, security, and energy efficiency. This literature study reviews key research articles on IoT and smart home apps, exploring their architectures, functionalities, and challenges.

#### **3.1.1 Smart Home Applications Based on IoT**

Smart home applications leverage IoT technology to provide interconnected and automated control over home devices. These apps enable users to manage devices such as lights, thermostats, security cameras, and appliances through a unified interface. According to Perera et al.[14], smart home apps significantly enhance user convenience by allowing remote control and automation of home devices. The study highlights the importance of interoperability and standardization in ensuring seamless integration of various IoT devices within smart home environments.

#### **3.1.2 System Architecture and Software for IoT-Based Smart Homes**

This typically features a cloud-based architecture that supports data storage, processing, and communication between devices. The architecture often includes components such as sensors, actuators, gateways, and cloud services. Research by Al-Fuqaha et al.[15] provides an in-depth analysis of system architectures for smart homes, emphasizing the need for robust data acquisition and processing software. The study categorizes software into three classes: operating systems, occupant tracking systems, and data acquisition systems, each playing a crucial role in the functionality of smart homes.

#### **3.1.3 Challenges and Solutions in Smart Home IoT Implementations**

While IoT and smart home technologies offer numerous benefits, they also present significant challenges. Security and privacy are primary concerns, as highlighted by Sicari et al [4].The study addresses several security threats, including unauthorized access, data breaches, and malware attacks, and suggests solutions such as encryption, secure communication protocols, and user authentication mechanisms. Furthermore, the research highlights the significance of user awareness and education in reducing security risks. [4].

#### **3.1.4 Forensic Readiness in “IoT and Smart Home Environments”**

In “IoT and smart home environments”, maintaining forensic readiness is crucial to ensure the effective collection and analysis of digital evidence during security incidents.



A study by Zawoad and Hasan [9] introduces the Forensic Readiness Framework for IoT (FRIoT), which outlines guidelines for implementing logging, data retention, and secure communication protocols within IoT ecosystems. This framework is designed to enhance the ability of organizations to respond to forensic investigations efficiently [9].

### **3.1.5 Prototypes and Tools for Forensic Readiness**

Several prototypes and tools have been developed to improve forensic readiness in IoT and smart home environments. An example is the IoT-Forensics Toolkit, which includes tools for capturing and analyzing data from various IoT devices. This toolkit addresses the unique challenges posed by the heterogeneity and resource constraints of IoT ecosystems. Additionally, the Smart Home Forensic Readiness Framework (SH-FRF) provides best practices for ensuring that smart home systems are forensically ready, including detailed logging of device interactions and secure data storage [16].

## **3.2 Digital Forensics and Digital Forensics readiness**

### **3.2.1 Digital Forensics**

Digital forensics is a field of forensic science focused on the identification, acquisition, preservation, analysis, and presentation of digital evidence. This field is utilized in numerous contexts, such as criminal investigations, corporate security, and incident response, to uncover and analyze data from digital devices and networks. Key processes in digital forensics include identification, acquisition, preservation, analysis, and presentation. Identification involves determining potential sources of digital evidence and understanding the scope of the investigation [5]. Acquisition refers to collecting digital evidence to ensure its integrity and relevancy in legal proceedings [5]. Preservation guarantees that digital evidence stays intact and safeguarded against tampering throughout the investigation [17]. Analysis involves examining the digital evidence to extract relevant information, identify patterns, and interpret findings [18]. Finally, presentation involves compiling and presenting the findings in a clear, concise manner suitable for legal or organizational proceedings [19]. Digital forensics involves various tools and techniques to handle different types of digital media, including computers, mobile devices, networks, and IoT devices. The growing complexity of digital environments necessitates continuous advancements in forensic methodologies and technologies to keep pace with evolving threats and technological advancements [20].

### **3.2.2 Digital Forensic Readiness:**

Digital forensic readiness involves the proactive measures and strategies organizations adopt to ensure they are prepared to conduct digital forensic investigations effectively when required. This concept emphasizes the importance of being prepared to respond to security incidents, data breaches, or legal inquiries involving digital evidence. Key components of digital forensic readiness include policies and procedures, training and awareness, technical infrastructure, data management, and legal compliance.

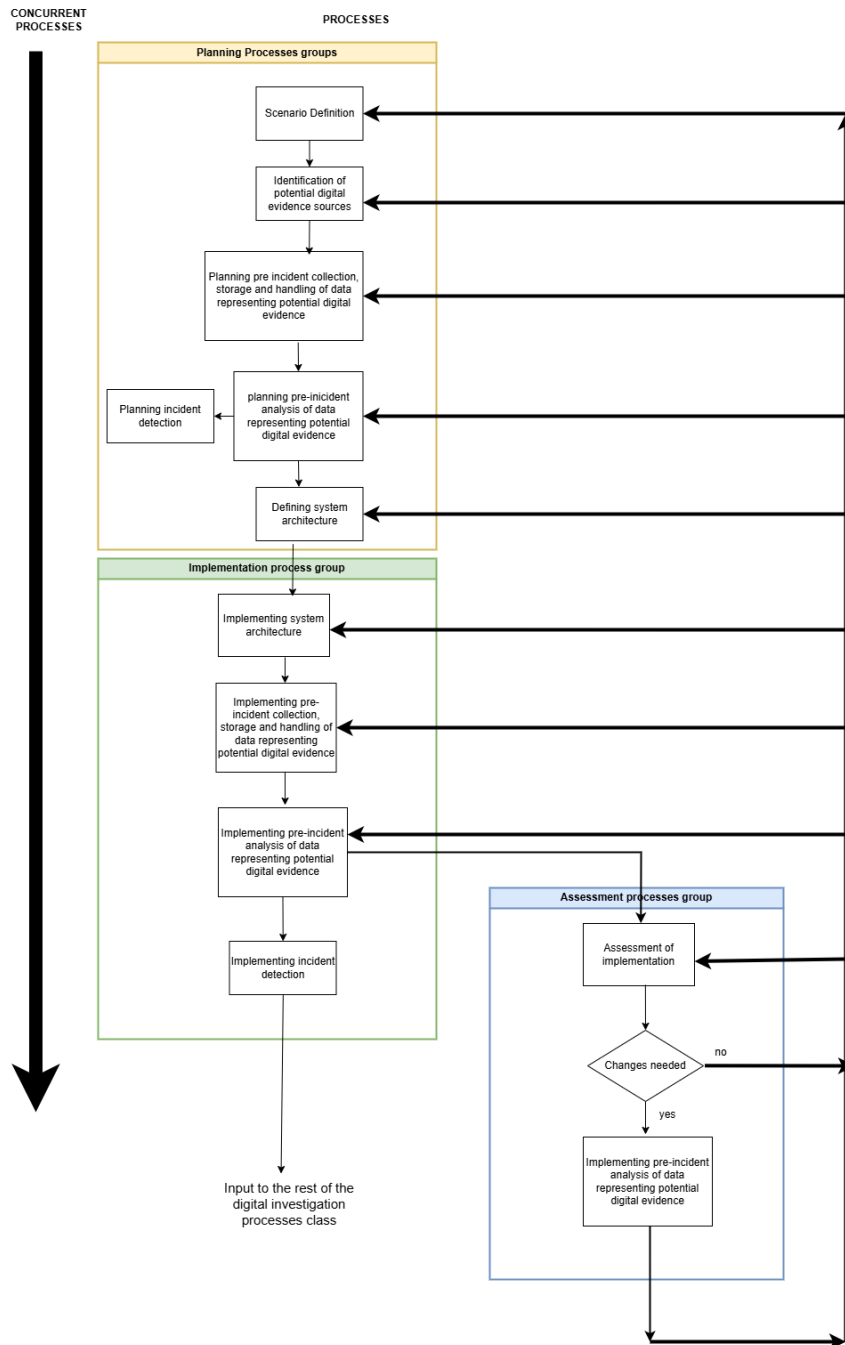
Key Components of Digital Forensic Readiness:

1. Establishing clear policies and procedures for the handling, storage, and analysis of digital evidence is crucial. This includes defining roles and responsibilities for forensic investigations [21].
2. Providing training and awareness programs for staff helps ensure that they understand the importance of digital forensics and the correct procedures for handling potential evidence [22].
3. Implementing technical solutions and tools that facilitate the collection, preservation, and analysis of digital evidence, such as forensic software and secure storage solutions, is also essential [20].
4. Proper data management practices, including data classification, retention, and disposal, make relevant information readily accessible for forensic analysis [23].
5. Finally, adhering to relevant legal and regulatory requirements ensures the admissibility of digital evidence in legal proceedings, including understanding jurisdictional differences and maintaining up-to-date knowledge of applicable laws

Digital forensic readiness is crucial for organizations to efficiently and effectively respond to digital forensic investigations, ensuring that they can quickly and accurately collect, preserve, and analyze digital evidence in the event of security incidents, data breaches, or legal inquiries [6].

### 3.2.3 ISO/IEC 27043

The ISO/IEC 27000 series of standards, which are designed to help organizations protect their information assets and manage information security risks. One notable one and perhaps more relevant to this study is the ISO/IEC 27043:2015. The readiness processes outlined in ISO/IEC 27043 provide a structured approach to preparing for potential digital incidents, ensuring that organizations are equipped to collect and analyze digital evidence effectively. Scenario definition serves as the starting point, where organizations identify various potential incident scenarios they might encounter, thus allowing them to tailor their forensic readiness strategies accordingly. Following this, the identification of potential digital evidence sources is crucial, as it enables organizations to pinpoint the specific data repositories and systems that could contain vital evidence during an incident. This is complemented by planning pre-incident collection and analysis, which includes defining procedures for gathering and analyzing data prior to any incidents occurring, ensuring that when an incident arises, the organization can quickly mobilize resources to gather relevant evidence. Additionally, the processes include planning for incident detection, which involves establishing monitoring systems that can identify anomalies or breaches in real-time[24].



**Fig. 2:** Readiness processes in ISO/IEC27043

The Digital Readiness Processes diagram in Figure 2 provides a visual outline of the systematic stages required for implementing a digital forensic readiness framework as explained above.

Once the preparatory planning is in place, organizations must focus on the practical implementation of these strategies. This begins with defining and implementing a robust system architecture that facilitates the collection, storage, and analysis of potential digital evidence. Effective storage and handling protocols are essential to maintain the integrity and security of the evidence collected, ensuring that it remains admissible in investigations. The implementation of pre-incident collection and analysis activities further enhances an organization's readiness, allowing for the systematic acquisition and examination of data that could be crucial during an incident. Furthermore, the processes emphasize the importance of ongoing assessment and refinement. Organizations must regularly assess the effectiveness of their forensic readiness measures and implement the results of these assessments to adapt to evolving threats, ensuring that their response capabilities remain robust and relevant in a dynamic digital landscape[24].

While ISO/IEC 27043 offers a robust framework for incident investigation, implementing the standard can present challenges. Organizations may need to invest in training, tools, and resources to build and maintain effective incident response and investigation capabilities. Additionally, the standard's emphasis on documentation and process adherence can be resource-intensive, requiring ongoing commitment and support from senior management[25]. Studies like the one we conducted in this research can help mitigate the limitations that the standard has.

### **3.3 Existing Digital forensic readiness techniques that can be applied to Apple's home**

#### **3.3.1 Live Forensics**

Live forensic is relevant to this research because the we employed a similar technique using SecureCollect hence in this section we explore existing tools that are utilized for live forensics.

Live forensics involves the collection and analysis of digital evidence from a system that is actively powered on and operating. This method is essential for capturing transient data that would be lost if the system were shut down. One key technique in live forensics is memory dumping, which involves capturing data within the RAM of a system to analyze active processes and connections, and other transient data. Tools like Volatility and FTK Imager are commonly used for memory dumping [26].

Another important technique is network traffic analysis, which involves monitoring and analyzing network traffic in real-time to identify suspicious activity and communication patterns. Wireshark and Zeek are popular tools for this purpose [27].

Additionally, process monitoring, which involves observing and recording active processes and their interactions, helps in understanding the behavior of malware or unauthorized programs. Sysinternals Process Monitor is an effective tool for process monitoring [28].

Log analysis, which entails collecting and analyzing log files from different sources, is also crucial. Centralized logging solutions like Splunk or ELK Stack [29] are often employed for comprehensive log analysis.

### 3.3.2 Digital Forensic Readiness Techniques

This involves preparing a company's systems and processes to facilitate efficient and effective forensic investigations. Comprehensive logging is a fundamental technique, as it ensures that relevant events are recorded across all systems and applications. This includes logs for user actions, system events, network activities, and application usage, which are critical for forensic analysis [30].

Secure data storage is another essential component, involving techniques such as encryption, access controls, and secure backup solutions to protect the integrity and availability of digital evidence. Implementing strong encryption mechanisms safeguards sensitive data from unauthorized access, while regular backups prevent data loss and ensure that historical data is available for analysis [23].

Automated incident detection and response are crucial for forensic readiness, employing automated tools to detect and respond to security incidents in real-time. Automated incident detection and response platforms can help reduce the effect of security incidents and ensure that relevant data is preserved for forensic purposes [6].

Additionally, preparing a forensic toolkit that includes the necessary hardware and software tools for conducting investigations is crucial. This might include forensic imaging tools, data recovery software, and network analysis tools to facilitate the extraction, preservation, and analysis of digital evidence [17].

### 3.3.3 Prototypes for Digital Forensic Readiness

Several prototypes and frameworks have been developed to enhance digital forensic readiness, particularly in the context of IoT and smart home environments. One notable example is the Forensic Readiness Framework for IoT (FRIoT) already mentioned above which focuses on ensuring that IoT environments are prepared for forensic investigations. [9].

Another example is the IoT-Forensics Toolkit, a prototype toolkit designed to facilitate forensic investigations in IoT environments. It includes tools for capturing and analyzing data from various IoT devices, addressing the specific challenges posed by the resource constraints of IoT ecosystems [31].

The Smart Home Forensic Readiness Framework (SH-FRF) mentioned in the previous subsection is another framework, which specifically tailored for smart home environments, outlining best practices for ensuring that smart home systems are forensically ready. This includes detailed logging of device interactions, secure data storage, and the use of standardized forensic tools [16].

### 3.4 Digital forensics methodologies and challenges in IoT environments

The emergence of the Iot has also presented various security issues, threats, and attacks and these include surveillance, viruses, a possibility of large scale botnets within the Iot networks and Denial of Service (DoS) attacks[32]. These security issues necessitate the need for forensics methodology for investigating IoT-related crime or even unrelated IoT-related crime where the data gathered from these devices could be used as evidence in civil cases or other criminal activities[16].

Traditional forensics is a relatively developed field and standardized, evidence gathering in traditional forensics collects data from sources such as the hard drives, Random Access Memory, system logs or any peripheral storage[16]. After the data has been collected, the data is examined, then analysed from a technical and legal point of view and finally reported.

Digital forensics in IoT presents new challenges when it comes to acquisition due to a lot of factors, one being how and where the data is stored in these Iot devices, which necessitate the need for new methodologies and frameworks that detail ways to gather evidence in IoT. Multiple papers are being written on this subject with new methodologies being developed, some of these methodologies are the "blockchain-based decentralized efficient investigation framework for IoT digital forensics" explored in [33], another one is 1-2-3 Zones proposed by Oriwoh, Edewede, et al in [32].

This literature study has explored the current landscape of digital forensics, computer security, and IoT, identifying significant advancements and ongoing challenges in these fields. It has highlighted key frameworks, such as the Forensic Readiness Framework for IoT, tools for enhancing forensic readiness, and methodologies designed to address the unique constraints of IoT devices. Despite these advancements, gaps remain in areas such as standardization, interoperability, and access limitations. These insights provide a foundational understanding to guide further research in developing robust forensic readiness in IoT and smart home ecosystems.

## 4 Results

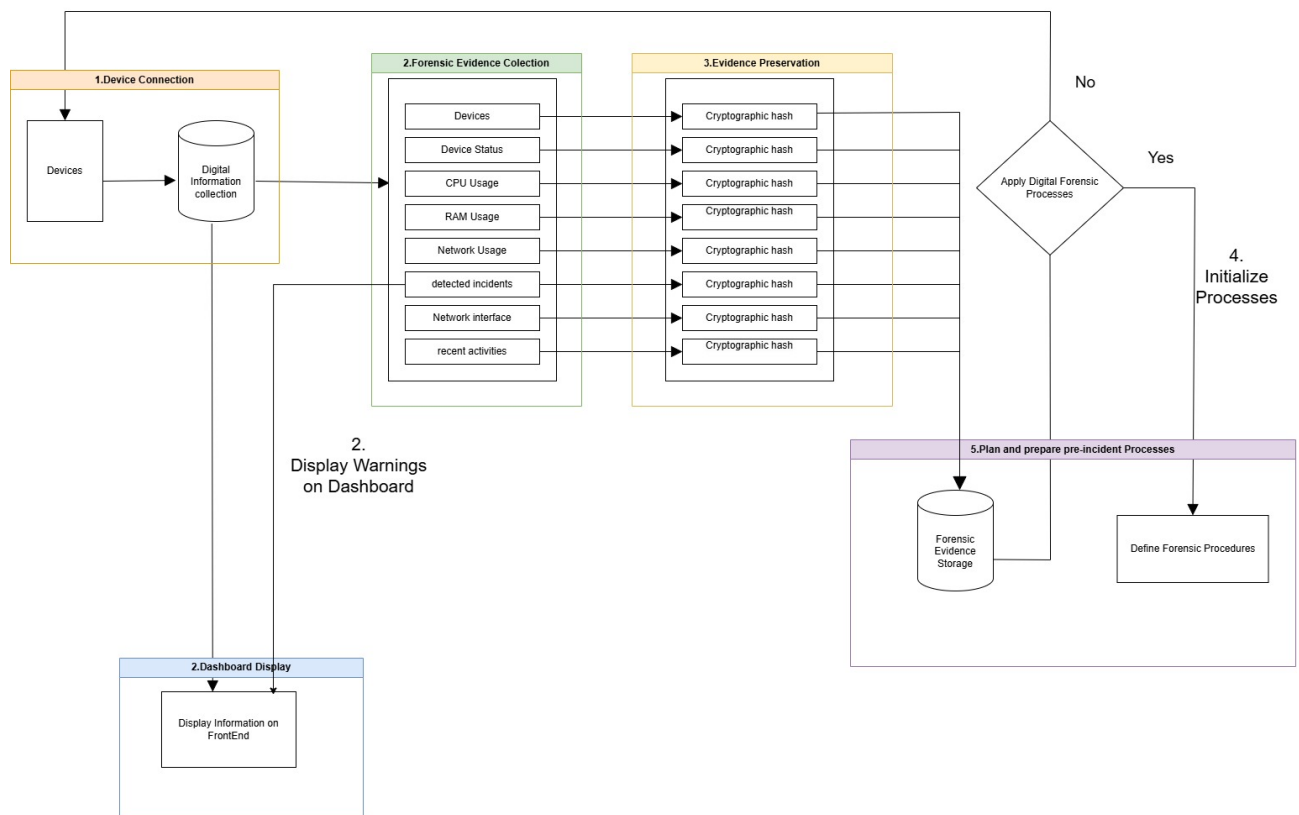
### 4.1 SecureCollect Design and Implementation

#### 4.1.1 Introduction

This chapter introduces the design and implementation of the forensic readiness prototype. SecureCollect is structured to operate in an Apple HomeKit/Homebridge ecosystem, enabling the secure collection and analysis of data from connected devices. The system's layered design integrates components such as Homebridge, smart devices, and the Home app, each facilitating forensic data collection.

#### 4.1.2 Details about the implementation

SecureCollect integrates several components that work together to enable forensic readiness in an Apple HomeKit/Homebridge environment. The flow of SecureCollect is shown in figure 3 below.

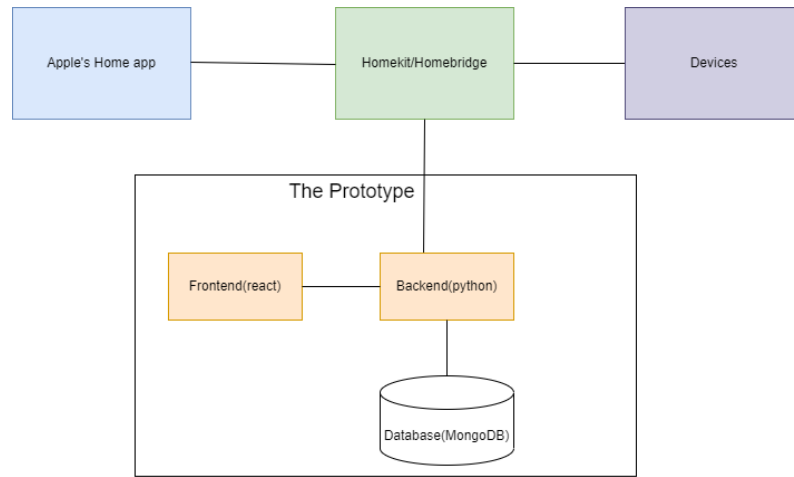


**Fig. 3:** SecureCollect Flow

### *Homebridge Interaction with Devices*

1. **Homebridge:** This layer serves as the bridge between Apple's Home app and various smart home devices. In the case of Homebridge, it extends compatibility to non-HomeKit-compatible devices, allowing them to function within the HomeKit ecosystem.
2. **Devices:** The connected devices, such as smart lights, locks, cameras, and other IoT devices, communicate with HomeKit or Homebridge. These devices generate logs and data that can be useful for forensic analysis, such as sensor readings, device states, and automation activities.
3. **Apple's Home App:** This app is the user interface where users control and monitor the devices through HomeKit or Homebridge.

Figure 4 shows an illustration of how Homebridge, Apple's home app and the physical devices interact with each other. The figure also shows SecureCollect's interaction with Homebridge, which it uses to collect potential evidential data through its APIs. SecureCollect's is further broken down into 3 components namely the frontend, backend and database



**Fig. 4:** The SecureCollect's interaction with devices, homebridge and Home App

### *SecureCollect - Frontend (React)*

The frontend of SecureCollect, built with React, serves as the user interface for our forensic readiness system. It allows users (e.g., investigators or administrators) to interact with the system, view data, and monitor the status of connected devices. The frontend is made up of the following components:

- a) **Login Component:** Ensures secure data retrieval through user authentication. It implements authentication mechanisms to safeguard access to device information and logs.



- b) DashBoard: It offers an overview of connected devices, recent activities, and system status (e.g., CPU, RAM, network status). Incorporates components like DeviceOverview, RecentActivities, and SystemStatus.
- c) Device Overview: A component within the dashboard that displays an overview of connected devices, retrieving data from the backend.
- d) Recent Activities and Logs: The frontend also fetches log information related to device interactions (e.g., turning lights on/off, door lock usage) and displays these activities.
- e) System Monitoring: This section displays health data such as CPU usage, memory, and network activity, allowing the system to detect potential issues like high resource usage or failures.
- f) DeviceDetails Page: Displays detailed information about devices, such as IP Addresses ,Device Names Firmware Versions and Ports.This page fetches information about devices and ensures users have an overview of relevant details.
- g) Displays specific details about the network interface, such as Interface Names ,IP Addresses, MAC Addresses, Port Numbers.It provides a comprehensive view of network interface details, assisting in understanding device connectivity.
- h) Logs Component: Fetches and displays logs from the backend. Displays both individual log entries and consolidated report views. Ensures real-time or historical log analysis with options for filtering and searching

Communication between the frontend and backend is conducted via API requests. This allows the frontend to request data (such as logs or device status) from the backend and then display it in a user-friendly manner.

### ***SecureCollect - Backend (Python)***

The backend, implemented using Python , is responsible for managing requests from the frontend and interacting with the Homebridge environment and the MongoDB database. Below is a list of the functionalities of the backend.

- a) Log Collection: The backend fetches logs from devices through Homebridge and stores them in the MongoDB database for future analysis. It can capture events like when a device is activated, modified, or encounters an issue.
- b) Device Monitoring: It also tracks the operational status of devices, collecting data on their performance and any warnings (e.g., high CPU usage). The backend sends this data to the frontend to be visualized in real time.

For secure communication between the frontend and backend, SSL certificates were generated using OpenSSL, which ensures encrypted data transfer.The SSLContext in Flask handles loading the certificate (cert.pem) and private key (key.pem).

Steps to Generate SSL Certificates:

**Generate a private key:** The private key is a critical part of asymmetric encryption. It is kept secret and only accessible to the entity it was generated for. This key is used to decrypt data that was encrypted with the associated public key. It also serves as a basis for creating digital signatures to verify identity or integrity. Figure 5 shows the command used to create the key. A key.pem file which stores the key is generated after the command is ran.

```

PS C:\Users\Test\Documents\prototype> openssl genrsa -out key.pem 2048
PS C:\Users\Test\Documents\prototype>

```

**Fig. 5:** Generating a private key

**Create a Certificate Signing Request (CSR):** Certificate Signing Request (CSR) is generated after the private key. The CSR is a request sent to a Certificate Authority (CA) to issue a certificate. It contains information like the organization name, domain, and public key, but it is not the certificate itself. Figure 6 shows the command and these details.

```

PS C:\Users\Test\Documents\prototype> openssl req -new -key key.pem -out cert.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ZA
State or Province Name (full name) [Some-State]:Gauteng
Locality Name (eg, city) []:Pretoria
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Prototype Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Shiluvelo
Email Address []:shiluvelo@shiluvelo.co.za

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345@prototype
An optional company name []:

```

**Fig. 6:** Creating Certificate Signing Request (CSR)

**Generate the SSL certificate:** Since this is for testing purposes, We generated our own self-signed SSL certificate using the csr file created previously , the key generated earlier is used to sign the key. The self-signed certificate is saved in the cert.pem file.. Figure 7 shows the command used to complete this process.

```

PS C:\Users\Test\Documents\prototype> openssl x509 -req -days 365 -in cert.csr -signkey key.pem -out cert.pem
Certificate request self-signature ok
subject=C=ZA, ST=Gauteng, L=Pretoria, O=Prototype Pty Ltd, CN=Shiluvulo, emailAddress=u19086352@tuks.co.za
PS C:\Users\Test\Documents\prototype>

```

**Fig. 7:** Generating the SSL certificate

Additionally, the backend hashes all the logs or data entries for integrity verification. This ensures that no data has been tampered with, which is essential for forensic readiness. Table 1 contains details about all the functionalities implemented and the libraries used for each of the component.

### ***Database (MongoDB)***

MongoDB serves as the database for SecureCollect. It stores log data, device configurations, system status updates, and any other information relevant to the forensic readiness of the system.

1. Log Storage: MongoDB contains historical data, storing logs retrieved from Homebridge. These logs include timestamped records of device interactions, errors, warnings, and status updates.
2. Efficient Retrieval: The backend queries MongoDB to retrieve logs, device status, and performance metrics, which are then sent to the frontend for visualization or used in report generation.
3. Integrity Verification: any log entries are hashed for integrity, MongoDB also stores these hashes, which can later be used to confirm that the logs haven't been altered.

### ***Components Integration***

All the components described above work together to form prototype and they help achieve the functions of SecureCollect as we intended for this research which is to implement digital forensic readiness processes. The integration of these components is summarized in the following list.

1. User Interaction: The user (e.g., forensic investigator) interacts with the system through the React-based frontend, which displays an overview of devices, logs, and system status.
2. Backend Operations: The frontend sends API requests to the Python backend, which processes these requests, retrieves data from the MongoDB database, and communicates with Homebridge to collect logs and status updates from devices.
3. Real-time Monitoring: The backend constantly monitors device and system performance, sending updated status information (e.g., CPU usage, memory usage) to the frontend for real-time display.
4. Log Analysis : Log data is stored in MongoDB. This process can help identify abnormal device behavior or security incidents.
5. Device Data: Devices communicate with Homebridge, sending operational data and logs to the system, which then processes them for forensic readiness purposes.

**Table 1:** Backend Architecture Overview (Rotated Table Layout)

Component	Description	Libraries/Tools Used
Flask Application	The main backend application that handles API requests, data retrieval, and device interactions.	Flask, Flask-CORS
Authentication	Ensures secure access to the system through login and token-based authentication.	bcrypt, werkzeug (for password hashing)
SSL Communication	Provides secure communication by using SSL certificates for data transmission over HTTPS.	ssl, custom SSLAdapter class, Requests library
MongoDB Database	Stores device data, hashed information, system statuses, logs, and recent activities.	Flask-PyMongo, bson, pymongo
Hashing	Ensures data integrity by hashing important data (e.g., device data, logs) using SHA-3 256-bit hashing for forensic purposes.	hashlib (for SHA-3 hashing)
Logging	Records detailed information about API calls, errors, and the general application status for monitoring and debugging.	Python's logging module
Device API	Retrieves device details (e.g., name, IP address, version) from connected HomeKit devices.	Requests, Custom SSLAdapter class for secure API communication
Network Interface API	Provides information about network interfaces connected to HomeKit devices.	Flask, SSLAdapter, Requests
System Status API	Retrieves system information such as CPU, RAM, and network status.	Requests, Flask
Logs API	Fetches logs from the local log file, hashes the logs, and stores them in the database.	hashlib, MongoDB
Recent Activities API	Retrieves recent device activities and stores hashed data and timestamps in MongoDB.	Requests, hashlib
SSL Certificate Generation	Enables secure SSL communication by generating self-signed SSL certificates for the application.	openssl (for certificate generation)

## 4.2 SecureCollect Experimentation

### 4.2.1 Introduction

This chapter will focus on conducting experimentation on SecureCollect to test its functionality, the security features implemented and how it adheres to digital forensic readiness processes as set out by the ISO27043 standard particularly the scenario definition, identification of potential digital evidence sources and the planning of pre-incident collection, storage and handling of data representing potential digital evidence, defining and implementing system architecture, implementation of pre-incident collection, storage and handling of data representing potential digital evidence[24].

### 4.2.2 Experiment Setup

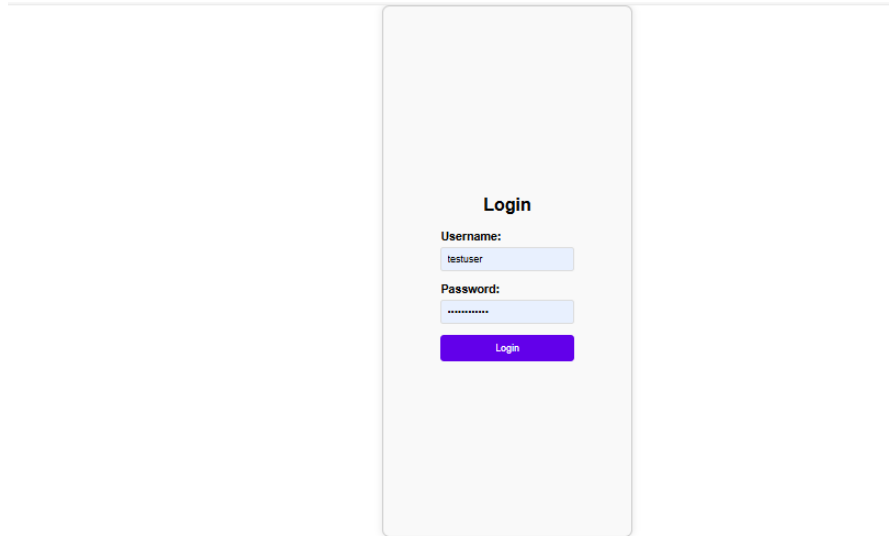
The previous chapter explains in details how SecureCollect is set for experimentation, so a reference to it will help understand how both the environment and SecureCollect are configured.

### 4.2.3 Experimentation Steps

This section of the experiment seeks to test the functionality of SecureCollect as a means to implement the forensic readiness processes, the following section outlines the test we took to experiment with the functionalities of SecureCollect.

#### ***Test 1: Testing authentication:***

When conducting the digital forensic readiness processes there is always an emphasis on the secure implementation of the processes for example, the collection of digital evidence must be done so securely, the transportation and storage must be done securely through the use of encryption, the preservation must be done securely through hashing e.t.c and as result SecureCollect has a log in page, which ensures that only an authenticated user(e.g an investigator) has access to SecureCollect. Details about the authenticated user i.e username and password are stored securely on MongoDB, and whenever a user needs to login, they are required to enter their details and their details are compared to what's already stored in the Database. The login page together with the data in the database are shown on figure 8a and 8b respectively.

A front-end login form titled "Login". It contains two input fields: "Username:" with the value "testuser" and "Password:" with masked characters "\*\*\*\*\*". Below the fields is a blue "Login" button.

(a) Devices on the front end.

```
QUERY RESULTS: 1-1 OF 1

_id: ObjectId('6707a032a90179577bbde99c')
username : "testuser"
password : "scrypt:32768:8:1$MkwUNTWmpKERzbjk$fe9016dc270089802cd56e9ce99f95d766..."
```

(b) Login Details stored in the Database

**Fig. 8:** User Authentication on the front end

In SecureCollect, secure evidence gathering is ensured by encrypting communication between the system and Homebridge using SSL certificates. These certificates (like cert.pem and key.pem) create a secure connection, protecting data transmission. Additionally, each API request made to or from Homebridge includes an authentication token in the request header to confirm the identity of the user. For example, the API request in the `getNetworkInterfaces()` function uses a token-based authentication scheme to authorize access and ensure that only authenticated requests can retrieve network , a demonstration of this is in figure 9 below.

```

@app.route('/api/network-interfaces', methods=['GET'])
def get_network_interfaces():
    ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
    ssl_context.load_cert_chain(certfile='cert.pem', keyfile='key.pem')

    session = requests.Session()
    session.mount('https://', SSLAdapter(ssl_context))

    url = 'http://localhost:8581/api/server/network-interfaces/system'
    headers = {'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImlNoakWx1dmVsbyIsIm5hbWUiOiJTaG1sdXZlbG81CjZG1pb1I6'}

    try:
        response = session.get(url, headers=headers)

```

Fig. 9: Authentication token

### Test 2: Testing Device Communication:

To get an overview of the functionality of SecureCollect, we developed a dashboard, that contains details about the system status, recent activities, device overview and links to other components that displays other data that is being collected, which can be seen on figure 10 below.

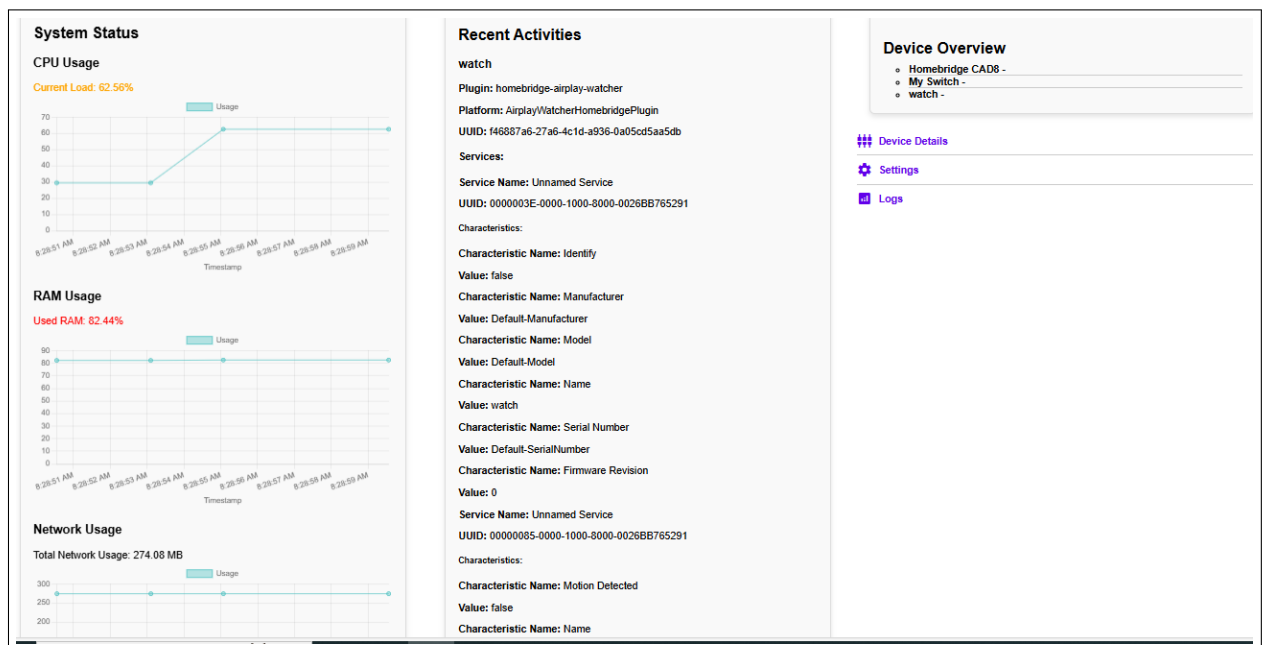
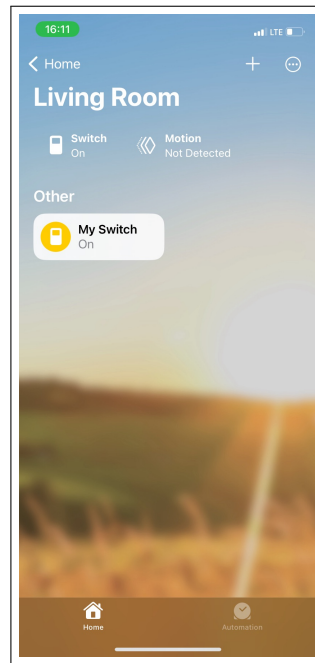


Fig. 10: Dashboard

The first thing before testing the communication of the devices is logging the type of devices connected to the home app at a particular time. Figure 11 shows a picture of the devices connected to apple's home, home app allows users to view and

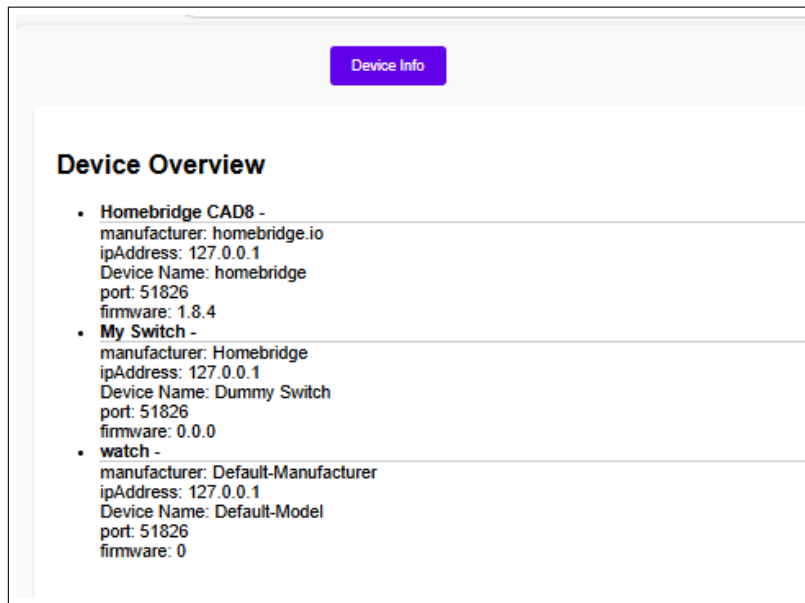
interact with these connected devices.



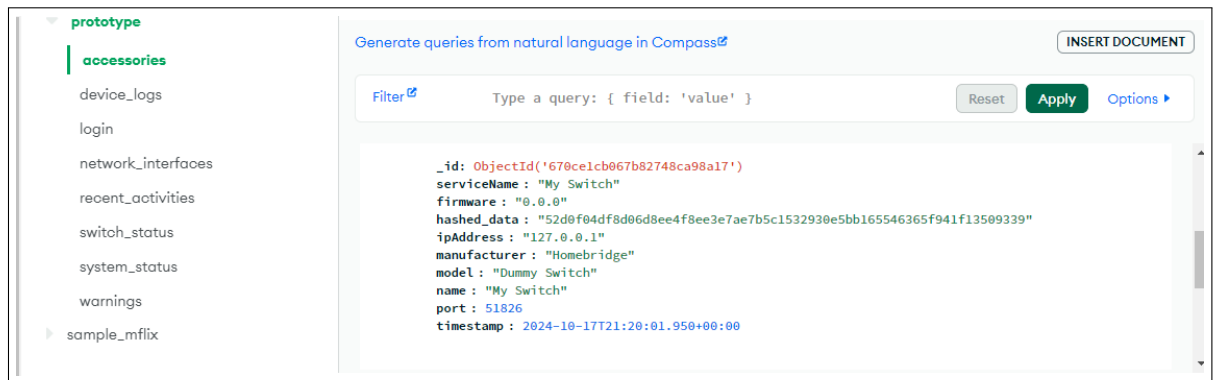
**Fig. 11:** Devices on Home app.

The device overview section of the dashboard shows the names of the devices currently connected to home app while the Device-details will show full details of the devices including names, IP addresses, device models, firmware and the ports the device use, this shown in figure 12a, next to it is figure 12b which shows how these devices are stored in the database, every stored devices is timestamped and also hashed to preserve its integrity.





(a) Devices on the front end.



(b) Devices stored on MongoDB

**Fig. 12:** Devices stored on MongoDB

In addition to the devices being identified, Details about the network interface of the system(this could be the device in which the home app is installed) are also collected, these details are shown in figure 13a, a corresponding image which shows the stored data is also attached shown in figure 13b.

- **Interface 2**

**ID:** 6711643286575492090f9e69

**Interface:** Wi-Fi

**Interface Name:** Intel(R) Wireless-N 7260

**MAC Address:** e8:b1:fc:ac:55:58

**IPv4 Address:** 10.32.156.31

**IPv4 Subnet:** 255.255.0.0

**IPv6 Address:** fe80::d32d:690b:c624:7980

**IPv6 Subnet:** ffff:ffff:ffff:ffff::

**DNS Suffix:** up.ac.za

**Operational State:** up

**Connection Type:** wireless

**Speed:** 72.2 Mbps

**DHCP:** Enabled

**Default Interface:** Yes

**Virtual Interface:** No

(a) Network Interface as seen from the front end

The screenshot shows the MongoDB Compass interface. On the left, the 'Project 0' sidebar lists collections under the 'prototype' database, with 'network\_interfaces' highlighted. The main panel shows a JSON document for a network interface:

```

{
  "_id": ObjectId('671176287f07143b72e0b4f1'),
  "iface": "Wi-Fi",
  "ifaceName": "Intel(R) Wireless-N 7260",
  "default": true,
  "ip4": "10.32.156.31",
  "ip4subnet": "255.255.0.0",
  "ip6": "fe80::d32d:690b:c624:7980",
  "ip6subnet": "ffff:ffff:ffff:ffff::",
  "mac": "e8:b1:fc:ac:55:58",
  "internal": false,
  "virtual": false,
  "operstate": "up"
}

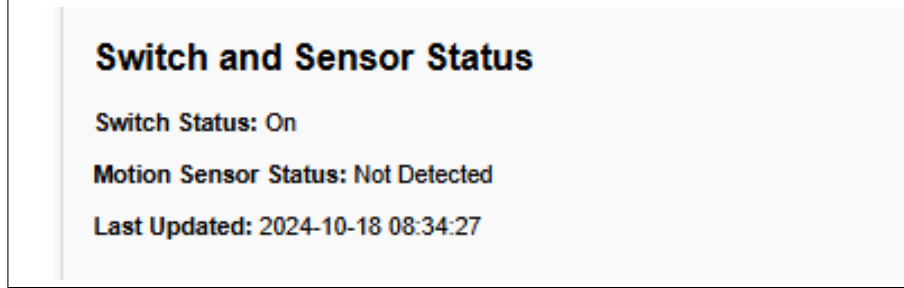
```

(b) Network Interface information stored on MongoDB

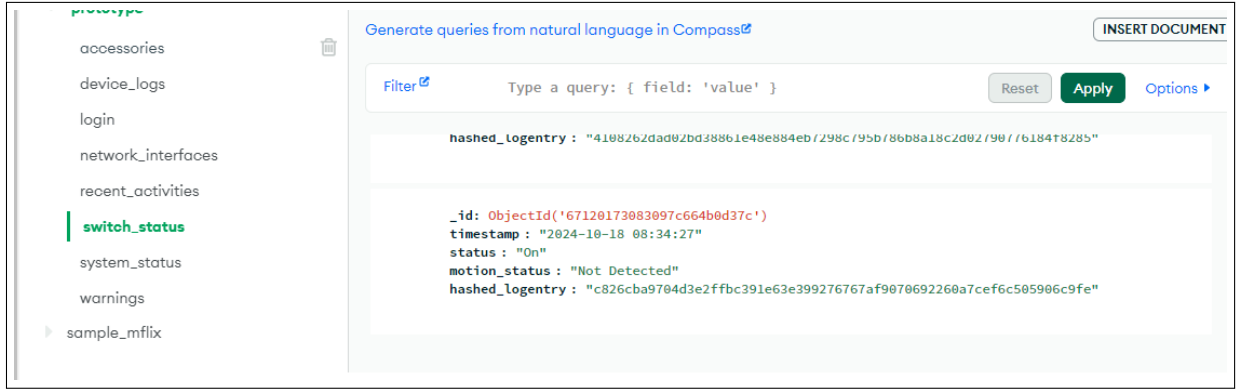
**Fig. 13:** Network Interface

### Test 3: Logging the devices status

We used Homebridge-compatible devices, a switch and motion sensor to interact with the Homebridge server, With the goal of Verifying that device statuses (e.g., on/off for the switch and motion detected/motion not detected) are correctly logged and stored in MongoDB using the /api/device/status endpoint. With the aim of Checking if the device logs, including actions such as "device on," "device off," are consistently captured and stored. Both the UI and MongoDB document are represented in figure 14.



(a) Device status on front end



(b) Device status stored in Database

Fig. 14: Device status

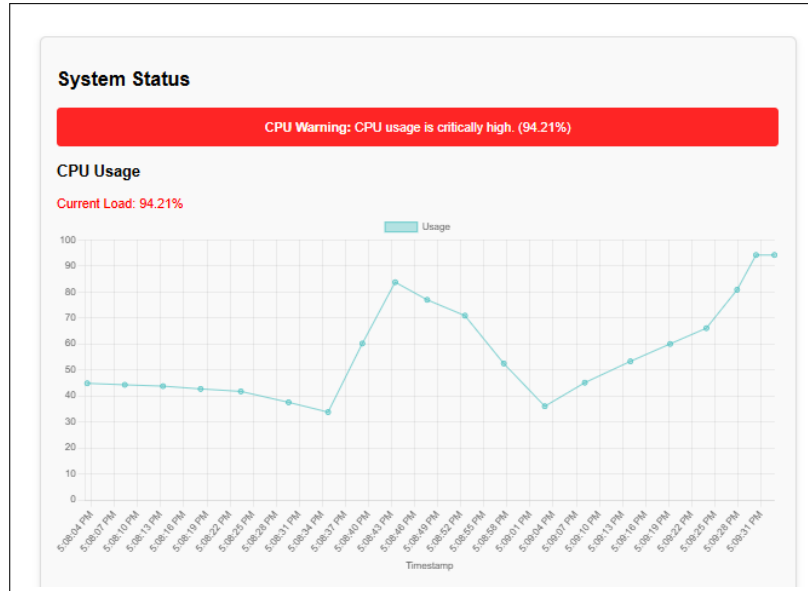
### Test 4: Testing system status and incident detection capability

As already shown on the Dashboard screenshot in figure 10, the system status component contains details about the CPU , RAM and Network usage, this data is continuously fetched to prepare for potential attacks such as denial of service attacks. The graphs in the component correspond to the usage in real-time, so if there is a sudden hike in usage, it will be seen on the graph and warnings will be issued on the UI, first of these warning is just a color coded health status, where green means

the usage is at a healthy level, orange is an early warning, then red is an adverse warning, and if the usage keeps rising then alerts will be issued to show critical level usage which could be a sign of a possible attack. The critical level warning is logged in the database for later reference when investigating the potential attack.

To test SecureCollect's response to an attack, a script was written in python to simulate a possible Denial of service attack, which works by continuously sending multiple requests to toggle a smart switch on and off using a large number of threads. The rapid succession of requests simulates a DoS attack that aims to overwhelm the system by sending a large volume of traffic in a short time, potentially disrupting its normal operation, the expected response of the system is that when the CPU usage reaches a certain usage threshold, then it should issue alerts on the front end , while saving these warnings in the database with timestamps.

The response to the simulated DoS attack is shown in figure 15, which shows a sudden spike in the graph at the exact time as the attack script is ran, and a warning about the rising CPU usage also pops out.



**Fig. 15:** CPU usage graph and warnings when the attack script is run.

Figure 16 shows an endpoint being called successfully to send these warnings to the database for safekeeping, enabling investigators to reference them during investigations. (The warning will continue to be stored until CPU usage returns below the threshold.)

```

127.0.0.1 - - [18/Oct/2024 17:01:30] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:30] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:32] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:32] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:33] "OPTIONS /api/warnings HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:33] "OPTIONS /api/warnings HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:33] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:33] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:35] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:35] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:35] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:35] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:37] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:37] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:38] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:38] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:41] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:41] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:41] "OPTIONS /api/warnings HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:41] "OPTIONS /api/warnings HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:41] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:41] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:43] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:43] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:44] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:44] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:47] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:47] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:47] "OPTIONS /api/warnings HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:47] "OPTIONS /api/warnings HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:47] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:47] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:49] "GET /api/system-status HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:49] "GET /api/system-status HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2024 17:01:50] "POST /api/warnings HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [18/Oct/2024 17:01:50] "POST /api/warnings HTTP/1.1" 201 -
127.0.0.1 - - [18/Oct/2024 17:01:51] "GET /api/system-status HTTP/1.1" 200 -

```

Fig. 16: Immediate logging of the warnings.

Figure 17 shows how the warnings are saved in MongoDB, each with a timestamp to allow investigators to determine the exact timing of each attack.

```

{
  _id: ObjectId('67127995e3c2a252effd5c49'),
  message: 'CPU usage is critically high.',
  metric: 'CPU',
  value: 92.48878923766816,
  timestamp: ISODate('2024-10-18T17:07:01.313Z')
},
{
  _id: ObjectId('67127998e3c2a252effd5c4b'),
  message: 'CPU usage is critically high.',
  metric: 'CPU',
  value: 92.48878923766816,
  timestamp: ISODate('2024-10-18T17:07:04.131Z')
},
{
  _id: ObjectId('67127a28e3c2a252effd5c6b'),
  message: 'CPU usage is critically high.',
  metric: 'CPU',
  value: 94.21233539891557,
  timestamp: ISODate('2024-10-18T17:09:28.496Z')
},
{
  _id: ObjectId('67127a2ae3c2a252effd5c6d'),
  message: 'CPU usage is critically high.',
  metric: 'CPU',
  value: 94.21233539891557,
  timestamp: ISODate('2024-10-18T17:09:30.313Z')
},
{
  _id: ObjectId('67127a2de3c2a252effd5c6f'),
  message: 'CPU usage is critically high.',
  metric: 'CPU',
  value: 94.21233539891557,
  timestamp: ISODate('2024-10-18T17:09:33.148Z')
}

```

Fig. 17: MongoDB storage of warnings, each with a unique timestamp.

#### 4.2.4 Challenges Faced

In this subsection we detail the challenges faced during the experimentation.

##### *Resource Limitations*

One significant limitation was the absence of physical HomeKit-compatible devices for testing. SecureCollect was tested in a simulated environment, which, while useful, could not fully replicate the behavior of real-world devices. Physical devices could have introduced more realistic network loads, device communication latencies, and potential hardware-specific issues (e.g., device overheating or network interface constraints) that would affect forensic data collection and system performance. The absence of these factors limited the scope of testing, particularly when simulating high-traffic events like DoS attacks, where physical devices might have responded differently compared to virtual devices.

### ***Handling Large Data Volumes***

The volume of data generated during extended tests posed challenges for storage management. MongoDB, while effective, occasionally encountered issues with large datasets. The data retrieval speed impacted the system's forensic readiness, highlighting the importance of optimizing database queries and storage strategies.

The experimentation phase demonstrated that SecureCollect successfully implements key aspects of digital forensic readiness, including secure logging, real-time monitoring, and incident detection. SecureCollect was able to capture relevant forensic data, store it securely, and issue alerts in response to potential attacks. The results indicate that while SecureCollect adheres to forensic readiness processes in theory, practical deployment in IoT environments requires additional optimization to handle real-world traffic, resource limitations, and data scalability. Future improvements could focus on enhancing network robustness, optimizing database performance, and improving the system's scalability for handling larger data volumes and more complex attacks.

## 5 Analysis and Evaluation

### 5.1 Introduction

This section critically evaluates the processes involved in this research using the ISO/IEC 27043 standard, focusing on digital forensic readiness in IoT through Apple’s Home app and Homebridge-enabled devices.

### 5.2 Scenario Definition

In this research, the scenario revolves around an IoT environment where Homebridge simulates HomeKit-compatible devices, enabling the evaluation of digital forensic readiness. The Home app is central to this ecosystem, and the primary challenge lies in ensuring the seamless integration of these devices for forensic purposes. Given that many non-Apple-certified devices are used, their lack of native support for the HomeKit protocol creates a complex digital forensic scenario. For instance, a potential security breach involving these non-HomeKit devices presents difficulties in evidence collection, due to the mixed nature of native and third-party devices.

### 5.3 Identification of Potential Digital Evidence Sources and Pre-Incident Collection Planning

The Home app, along with Homebridge-enabled devices, generates various forms of digital evidence, including device activity logs, user interactions, and network communication data. The identification process focused on these elements as key sources of digital evidence. Planning for pre-incident collection involved developing SecureCollect that allowed for continuous logging, device configuration monitoring, and network traffic analysis, ensuring that relevant data could be collected before any incident occurs. By proactively storing these logs, the research prepared for the potential need to extract data in the event of a forensic investigation.

### 5.4 Storage and Handling of Data Representing Potential Digital Evidence

The collected data was stored securely using MongoDB, which was chosen for its ability to efficiently manage large datasets typically generated by IoT devices. The research emphasized the importance of encryption to preserve data integrity and confidentiality, especially considering the sensitive nature of smart home data. Additionally, secure handling mechanisms were implemented in SecureCollect, such as access control and encryption during transmission, ensuring that only authorized personnel could access the data. This process aligns with ISO/IEC 27043’s guidelines on preserving the integrity of potential evidence throughout the collection and handling phases.

### 5.5 Defining and Implementing System Architecture

The architecture of the system was designed to mimic a real-world IoT environment. The Homebridge platform was used to connect non-HomeKit-compatible devices,



while the Home app acted as the control center. The system's architecture was designed with forensic readiness in mind, incorporating continuous data logging, secure storage, and the ability to simulate device interactions. However, the research faced limitations in replicating the full range of devices typically found in an IoT environment due to the lack of physical devices, potentially affecting the comprehensiveness of the forensic analysis.

## **5.6 Implementation of Pre-Incident Collection**

Pre-incident collection was achieved through the proactive logging of device interactions and network traffic. The research utilized the Homebridge logs and network monitoring tools to capture communication data between devices. These logs were stored in a structured format, ensuring they could be easily retrieved and analyzed. The continuous logging approach ensured that in the event of an incident, forensic investigators could rely on previously collected data, enhancing the system's forensic readiness.

## **5.7 Storage and Handling of Data Representing Potential Digital Evidence (Post-Incident)**

Post-incident, the system allowed for the retrieval of stored logs and network data. The handling of this data followed strict guidelines to prevent tampering or corruption. By using hashing techniques to verify data integrity, the research ensured that forensic evidence could be used in legal proceedings if necessary. This process adhered to the principles of ISO/IEC 27043, particularly regarding maintaining a chain of custody and ensuring data remains unchanged during analysis.

As shown above the SecureCollect complies with the readiness processes of the ISO/IEC 27043 standard. T. Janarthanan et al [16] Proposes a Iot forensic as a service, SecureCollect is good step in the right direction towards that, cloud service providers could use SecureCollect implemented in this research to build such services which can make evidence gathering and consequently digital forensic investigations a lot more efficient.

# **6 Chapter 6: Conclusion**

## **6.1 Introduction**

This concluding chapter reflects on the study's objectives, revisiting the problem statement and research questions addressed through the development of SecureCollect for the Apple HomeKit environment. The chapter summarizes how each research question was approached, highlighting the methods used to identify, secure, and preserve data for forensic purposes. Additionally, it explores the broader implications of these findings, acknowledging the study's limitations and suggesting directions for future research to enhance forensic readiness across diverse IoT ecosystems.

## 6.2 Problem Statement and Research Questions Revisited

The research questions were designed to guide the exploration of the problem and were centered around three key areas: how data can be identified and collected in HomeKit environments, how data can be securely stored and preserved for forensic analysis, and how this data can be utilized to support investigations. Through the development of SecureCollect, this study effectively addressed these questions by making use of methods for identifying critical forensic data, leveraging secure storage solutions, and ensuring the integrity of data collection through hashing and logging techniques.

The research demonstrated that forensic readiness in HomeKit ecosystems is not only feasible but also necessary to ensure that critical forensic data is available when needed. SecureCollect provides a systematic approach to identification, collecting, preserving, and utilizing IoT data, meeting the objectives of the problem statement. Key components, such as the integration of logging mechanisms, hashing for integrity, and a focus on proactive forensic data collection, ensure that the HomeKit environment can support future forensic investigations.

## 6.3 Future Research

However, the research also uncovered areas for further investigation. One limitation is the proprietary nature of Apple’s HomeKit, which restricts access to certain internal data structures and processes. Future research could focus on collaborating with Apple to open up APIs for better forensic support another avenue for expansion would be testing this SecureCollect across multiple IoT ecosystems, allowing for a comparative study of different smart home platforms and their forensic capabilities.

## References

- [1] Neha Sharma, Madhavi Shamkuwar, and Inderjit Singh. “The history, present and future with IoT”. In: Internet of things and big data analytics for smart generation (2019), pp. 27–51.
- [2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The internet of things: A survey”. In: Computer networks 54.15 (2010), pp. 2787–2805.
- [3] Yuchen Yang et al. “A survey on security and privacy issues in Internet-of-Things”. In: IEEE Internet of things Journal 4.5 (2017), pp. 1250–1258.
- [4] Sabrina Sicari et al. “Security, privacy and trust in Internet of Things: The road ahead”. In: Computer networks 76 (2015), pp. 146–164.
- [5] Keyun Ruan et al. “Cloud forensics”. In: Advances in Digital Forensics VII: 7th IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL, USA, 2011. Springer, 2011, pp. 35–46.
- [6] Karen Kent, Suzanne Chevalier, and Tim Grance. “Guide to integrating forensic techniques into incident”. In: Tech. Rep. 800-86 (2006).
- [7] Wei Zhou et al. “Discovering and understanding the security hazards in the interactions between {IoT} devices, mobile apps, and clouds on smart home platforms”. In: 28th USENIX security symposium (USENIX security 19). 2019, pp. 1133–1150.
- [8] Tam Thanh Doan et al. “Towards a resilient smart home”. In: Proceedings of the 2018 workshop on IoT security and privacy. 2018, pp. 15–21.
- [9] Shams Zawoad and Ragib Hasan. “Faiot: Towards building a forensics aware eco system for the internet of things”. In: 2015 IEEE International Conference on Services Computing. IEEE, 2015, pp. 279–284.
- [10] Nicholas E Fifield. “Apple HomeKit application and cost breakdown”. In: (2020).
- [11] Homebridge Contributors. Homebridge: Open-source software to support HomeKit integration. Accessed: 2024-10-23. 2024. URL: <https://homebridge.io/>.
- [12] Victor R Kebande et al. “Holistic digital forensic readiness framework for IoT-enabled organizations”. In: Forensic Science International: Reports 2 (2020), p. 100117.
- [13] Brij B Gupta and Megha Quamara. “An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols”. In: Concurrency and Computation: Practice and Experience 32.21 (2020), e4946.
- [14] Charith Perera et al. “Context aware computing for the internet of things: A survey”. In: IEEE communications surveys & tutorials 16.1 (2013), pp. 414–454.
- [15] Ala Al-Fuqaha et al. “Internet of things: A survey on enabling technologies, protocols, and applications”. In: IEEE communications surveys & tutorials 17.4 (2015), pp. 2347–2376.
- [16] T Janarthanan, M Bagheri, and S Zargari. “IoT forensics: an overview of the current issues and challenges”. In:

- Digital Forensic Investigation of Internet of Things (IoT) Devices (2021), pp. 223–254.
- [17] Eoghan Casey. Digital evidence and computer crime: Forensic science, computers, and the internet. Academic press, 2011.
  - [18] Bill Nelson, Amelia Phillips, and Christopher Steuart. Guide to computer forensics and investigations. Course Technology Cengage Learning, 2010.
  - [19] Mohammad Iftekhhar Husain, Ibrahim Baggili, and Ramalingam Sridhar. “A simple cost-effective framework for iPhone forensic analysis”. In: Digital Forensics and Cyber Crime: Second International ICST Conference, ICDF2C 2010, Abu Dhabi. Springer. 2011, pp. 27–37.
  - [20] Mark M Pollitt. “An ad hoc review of digital forensic models”. In: Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE’07). IEEE. 2007, pp. 43–54.
  - [21] Simson L Garfinkel. “Digital forensics research: The next 10 years”. In: digital investigation 7 (2010), S64–S73.
  - [22] Gary Palmer et al. “A road map for digital forensic research”. In: First digital forensic research workshop, utica, new york. 2001, pp. 27–30.
  - [23] Kevin D Mitnick and William L Simon. The art of intrusion: the real stories behind the exploits of hackers, intruders and deceivers. John Wiley & Sons, 2009.
  - [24] ISO/IEC 27043: Information technology – Security techniques – Incident investigation principles and Accessed: October 16, 2024. 2015. URL: <https://www.iso.org/standard/44407.html>.
  - [25] Aleksandar Valjarević, Hein Venter, and Ranko Petrović. “ISO/IEC 27043: 2015—Role and application”. In: 2016 24th Telecommunications Forum (TELFOR). IEEE. 2016, pp. 1–4.
  - [26] Michael Hale Ligh et al. The art of memory forensics: detecting malware and threats in windows, linux. John Wiley & Sons, 2014.
  - [27] Chris Sanders and Jason Smith. Applied network security monitoring: collection, detection, and analysis. Elsevier, 2013.
  - [28] Pavel Yosifovich et al. “System architecture, processes, threads, memory management, and more”. In: (No Title) (2017).
  - [29] Marcin Bajer. “Building an IoT data hub with Elasticsearch, Logstash and Kibana”. In: 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW). IEEE. 2017, pp. 63–68.
  - [30] Raffael Marty. Applied security visualization. Addison-Wesley Professional, 2008.
  - [31] Cyber Law. “Journal of Digital Forensics, Security and Law, Vol. 6 (3)”. In: (2011).
  - [32] Edewede Oriwoh et al. “Internet of things forensics: Challenges and approaches”. In: 9th IEEE International Conference on Collaborative computing: networking, Applications and Wo. IEEE. 2013, pp. 608–615.

- [33] Jung Hyun Ryu et al. “A blockchain-based decentralized efficient investigation framework for IoT digital forensics”. In: The Journal of Supercomputing 75 (2019), pp. 4372–4387.