UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF DIGITAL SYSTEMS

IoT Forensics

Doctoral Dissertation

Dragonas Evangelos

Piraeus, 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ψηφιακή εγκληματολογία στο Διαδίκτυο των Πραγμάτων

Διδακτορική Διατριβή

Δραγώνας Ευάγγελος

Πειραιάς, 2023

Advisory Committee

Costas Lambrinoudakis
Professor (Supervisor)
University of Piraeus

Lilian Mitrou,
Professor
University of the Aegean

Stefanos Gritzalis,
Professor
University of Piraeus

Piraeus, 2023

Approval Sheet

UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF DIGITAL SYSTEMS

This is to certify that the Dissertation presented by Dragonas Evangelos, entitled "***IoT Forensics***", submitted in fulfillment of the requirement for the degree of Doctor of Philosophy, complies with the regulation of the University of Piraeus and meets the accepted standards with respect to originality.

Piraeus, 2023

Dissertation Committee

Costas Lambrinoudakis
Professor (Supervisor)
University of Piraeus

Lilian Mitrou,
Professor
University of the Aegean

Stefanos Gritzalis,
Professor
University of Piraeus

Xenakis Christos,
Professor
University of Piraeus

Katsikas Sokratis,
Professor
Norwegian University of Science and Technology

Kalloniatis Christos,
Professor
University of the Aegean

Magkos Emmanouil (Manos),
Professor
Ionian University

Piraeus, 2023

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. I additionally declare that the opinions expressed in this document are my sole responsibility and do not necessarily represent the official position of the University of Piraeus.

Dragonas Evangelos
Piraeus, October, 2023

Dedicated to my beloved Stefania

and my treasured Stitch

“Education is not the filling of a pail, but the lighting of a fire”

William Butler Yeats

Ευχαριστίες (Acknowledgments)

Ολοκληρώνοντας το ταξίδι της εκπόνησης της διδακτορικής μου διατριβής, θα ήθελα να ευχαριστήσω προσωπικά εκείνους που με υποστήριξαν κατά τη διάρκεια αυτής της προσπάθειας και να εκφράσω την ειλικρινή μου ευγνωμοσύνη για την ανεκτίμητη συνεισφορά τους.

Πρώτα από όλους, θα ήθελα να ευχαριστήσω μέσα από τα βάθη της καρδιάς μου, τον επιβλέποντα μου κ. Κωσταντίνο Λαμπρινουδάκη, Καθηγητή του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστήμιου Πειραιώς, ο οποίος πιστεύοντας σε εμένα από την πρώτη στιγμή, με υποστήριξε καθ' όλη τη διάρκεια του ερευνητικού αυτού ταξιδιού. Μου προσέφερε μεταξύ άλλων όλους τους πόρους και τα μέσα που χρειάστηκαν, τις πολύτιμες γνώσεις και τον σημαντικό χρόνο του. Οι καίριες επισημάνσεις του σε συνδυασμό με τα προαναφερθέντα, συνέδραμαν καθοριστικά στην επιτυχημένη ολοκλήρωση της παρούσας διδακτορικής διατριβής. Τον εκτιμώ βαθύτατα ως άνθρωπο και ως επιστήμονα και θεωρώ ηθική μου ανταμοιβή, τη συνέχιση της συνεργασίας μας σε νέα ερευνητικά μονοπάτια.

Έπειτα, θα ήθελα να πω ένα μεγάλο ευχαριστώ στην κα. Λίλιαν Μήτρου, Καθηγήτρια στο Πανεπιστήμιο Αιγαίου και μέλους της τριμελούς συμβουλευτικής επιτροπής, για τις επιστημονικές συμβουλές της στην παρούσα διατριβή, για την πίστη της στο εγχείρημα μου και κυρίως διότι η συνεργασία μου με τον κ. Λαμπρινουδάκη ξεκίνησε αρχικά χάρη στη δική της έμπνευση. Επίσης, ευχαριστώ θερμά τον κ. Στέφανο Γκρίτζαλη, Καθηγητή στο Πανεπιστήμιο Πειραιώς και μέλους της τριμελούς συμβουλευτικής επιτροπής, για την παραγωγική κριτική και τα ενθαρρυντικά σχόλια του στα πλαίσια της παρούσας διατριβής, τα οποία αποτελούν κινητήρια δύναμη για εμένα και για την περαιτέρω συνέχιση της έρευνας μου.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω ιδιαίτερα και να εκφράσω την εκτίμηση μου σε όλα τα μέλη της επταμελούς εξεταστικής επιτροπής, ξεκινώντας με τον κ. Σωκράτη Κάτσικα, Καθηγητή στο Πανεπιστήμιο «Norwegian University of Science and Technology», τον κ. Χρήστο Ξενάκη, Καθηγητή στο Πανεπιστήμιο Πειραιώς, τον κ. Εμμανουήλ Μάγκο, Καθηγητή στο Ιόνιο Πανεπιστήμιο και τον κ. Χρήστο Καλλονιάτη, Καθηγητή στο Πανεπιστήμιο Αιγαίου, τόσο για την πρόθυμη συμμετοχή και ενεργή συνεισφορά τους στην κρίση της διδακτορικής μου διατριβής, όσο και για την εποικοδομητική κριτική και τις χρήσιμες συμβουλές που μοιράστηκαν μαζί μου στα πλαίσια αυτής.

Ακόμη, είμαι ευγνώμον και ευχαριστώ πολύ τον φίλο μου και αγαπητό συνάδελφο, Μιχάλη Κωτσή, για την εξαιρετικά σημαντική συνεισφορά του στην παρούσα διατριβή και τη στήριξη που μου παρείχε σε όλη αυτή την προσπάθεια.

Κλείνοντας, θα ήθελα να ευχαριστήσω τους γονείς μου και τα αδέρφια μου για τη συμπαράσταση και την υποστήριξη τους σε αυτό το ταξίδι. Επιπρόσθετα, θέλω να εκφράσω τις ευχαριστίες μου στον κ. Θανάση και στην κα. Σοφία για τις όμορφες συζητήσεις και τις απόψεις που ανταλλάξαμε στα πλαίσια της διατριβής αυτής. Τέλος, το μεγαλύτερο ευχαριστώ μου το οφείλω στην αγαπημένη μου σύζυγο Στεφανία και στον φίλο μου Στιτς, οι οποίοι βρίσκονταν πάντα εκεί για μένα, να αφουγκραστούν τις ανησυχίες μου, να με στηρίξουν σε κάθε αποτυχία και να γιορτάσουν μαζί μου κάθε επιτυχία στην πορεία αυτού του ταξιδιού.-

Με τιμή,

Ευάγγελος Δραγώνας

Abstract

The doctoral dissertation entitled “Internet of Things Forensics” focuses on the digital investigation (a.k.a. forensic analysis/examination/investigation) of a wide variety of devices that belong to the Internet of Things (IoT) realm.

This dissertation was primarily motivated by the observation that the IoT landscape is crowded with numerous active manufacturers with a diverse range of products. This heterogeneity complicates a comprehensive forensic analysis of the data stored within these devices, particularly when they are involved in criminal activity. Many reasons contribute to this complexity.

To begin with, each IoT device could store its data in a single location (e.g., a mobile companion app, cloud, etc.) or across multiple places. Secondly, the forensic software that an investigator has access to may not efficiently parse or not even access the data recorded by each IoT device. It is therefore evident that digital forensic examiners investigating a crime involving IoT products need to know where each IoT device saves its data, how to correctly interpret this information, avoiding misinterpretation, and be aware of the available tools that can assist with this assignment.

Considering the current number of IoT devices, accomplishing this task is almost impossible. As a consequence, forensic studies of IoT devices are invaluable. Their findings can serve as a reference for any investigator working on a case with similar appliances, or for digital forensic software companies seeking to update their products to better decode this information.

Therefore, this doctoral dissertation aims to shed light on selected IoT devices from a Digital Forensics perspective. To achieve this goal, forensic analysis of several IoT devices was carried out. The examined appliances were manufactured by leading companies in the IoT market and mainly fall under the categories of Security Systems and Closed-Circuit Television Surveillance Systems. During the course of this dissertation, new open-source digital forensic software was developed and existing software was updated based on the findings from the conducted research. By fulfilling the aforementioned goal, the intention is to aid investigators worldwide in their pursuit of justice and to assist digital forensic software companies in enhancing their products.

Field of Science: Digital Forensics

Keywords: Digital Forensics, Internet of Things, IoT, Smart Home, Security System, Motion Sensor, Opening Sensor, Closed-Circuit Television, CCTV, Surveillance System, Camera, Operating System, Forensic Software, XIAOMI, HIKVISION, DAHUA Technology, AJAX Systems

Περίληψη

Η διδακτορική διατριβή με θέμα «Ψηφιακή Εγκληματολογία στο Διαδίκτυο των Πραγμάτων» επικεντρώνεται στην εγκληματολογική εξέταση πλήθους συσκευών που ανήκουν στον ψηφιακό «κόσμο» του Διαδικτύου των Πραγμάτων (ΔτΠ).

Κίνητρο για την παρούσα διατριβή, αποτέλεσε πρωτίστως η παρατήρηση ότι στο ΔτΠ δραστηριοποιούνται πάρα πολλοί κατασκευαστές, με πλήθος διαφορετικών προϊόντων ο καθένας. Το γεγονός αυτό, καθιστά εξαιρετικά δύσκολη την αποτελεσματική ψηφιακή διερεύνηση των ανωτέρω συσκευών, σε περίπτωση που αυτές εμπλακούν με οποιοδήποτε τρόπο σε κάποιο έγκλημα. Η δυσκολία αυτή οφείλεται σε πολλούς παράγοντες.

Αρχικά, κάθε τέτοια συσκευή δύναται να αποθηκεύει τα δεδομένα της είτε σε ένα μέρος (π.χ. στη συνοδευτική εφαρμογή του τηλεφώνου, στο νέφος, κ.ά.) ή τμηματικά σε περισσότερα από ένα μέρη. Έπειτα, τα εγκληματολογικά λογισμικά που οι ερευνητές ψηφιακής εγκληματολογίας έχουν στη διάθεση τους, δυσκολεύονται να υποστηρίξουν την ανάλυση των δεδομένων των προϊόντων κάθε κατασκευαστή. Τέλος, οι ερευνητές που αναλαμβάνουν τη διερεύνηση ενός εγκλήματος στο οποίο εμπλέκονται συσκευές του ΔτΠ, χρειάζεται να γνωρίζουν που αποθηκεύονται τα δεδομένα των συσκευών που εξετάζουν, πώς να τα ερμηνεύσουν, πώς να μην τα παρερμηνεύσουν, καθώς επίσης πρέπει να γνωρίζουν και ποια λογισμικά μπορούν να τους βοηθήσουν στο έργο τους.

Εάν αναλογιστεί κανείς το μέγεθος που έχει σήμερα το ΔτΠ, αντιλαμβάνεται ότι κάτι τέτοιο είναι πρακτικά αδύνατο. Γι’ αυτό το λόγο, οι εγκληματολογικές μελέτες συσκευών του ΔτΠ είναι εξαιρετικής σημασίας, τα αποτελέσματα των οποίων μπορούν να χρησιμοποιηθούν ως σημείο αναφοράς είτε από τους ερευνητές που εξετάζουν μία υπόθεση με παρόμοιες συσκευές είτε από τις εταιρείες ανάπτυξης εγκληματολογικών λογισμικών που επιδιώκουν την ενημέρωση των προϊόντων τους, ώστε να αποκωδικοποιούν σωστά αυτά τα δεδομένα.

Επομένως, στόχος της παρούσας διατριβής αποτελεί η «χαρτογράφηση» ενός μέρους του ΔτΠ, υπό το πρίσμα της επιστήμης της Ψηφιακής Εγκληματολογίας. Για την επίτευξη του στόχου αυτού, πραγματοποιήθηκε εγκληματολογική εξέταση συσκευών του ΔτΠ που κατασκευάζονται από εταιρείες ηγέτες στον χώρο και οι οποίες ανήκουν ως επί το πλείστον στις ακόλουθες κατηγορίες προϊόντων: συστήματα ασφαλείας και συστήματα παρακολούθησης. Στο πλαίσιο της παρούσας διατριβής, αναπτύχθηκαν νέα εγκληματολογικά λογισμικά ανοιχτού κώδικα και ενημερώθηκαν ήδη υπάρχοντα με βάση τα ευρήματα της διενεργηθείσας έρευνας. Η εκπλήρωση του εν λόγω στόχου αποσκοπεί στην υποβοήθηση της αποστολής των ερευνητών ψηφιακής

εγκληματολογίας ανά τον κόσμο να υπηρετούν τη δικαιοσύνη και των εταιρειών εγκληματολογικών λογισμικών να βελτιώνουν τα προϊόντα τους.

Θεματική Περιοχή: Ψηφιακή Εγκληματολογία

Λέξεις Κλειδιά: Ψηφιακή Εγκληματολογία, Διαδίκτυο των Πραγμάτων, ΔτΠ, Έξυπνο Σπίτι, Σύστημα Ασφαλείας, Αισθητήρας Κίνησης, Αισθητήρας Ανοίγματος, Κλειστό Κύκλωμα Τηλεόρασης, Καταγραφικό Μηχάνημα, Κάμερα, Λειτουργικό Σύστημα, Εγκληματολογικό λογισμικό, XIAOMI, HIKVISION, DAHUA Technology, AJAX Systems

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
BLOF	Blockchain-Based Forensic Model for IoT
BPLIST	Binary Property List
BSAF	Blockchain-Assisted Shared Audit Framework
CCTV	Closed-Circuit Television
CSP	Cloud Service Provider
DF	Digital Forensics
DFIF-IoT	Digital Forensic Investigation Framework for IoT
DFIF-IoT	Digital Forensic Investigation Framework for IoT
DFIM	Digital Forensics Investigation Model for IoT
DFIR	Digital Forensics and Incident Response

DHFS	Dahua File System
DVR	Digital Video Recorder
FAIoT	Forensics-aware IoT
FFS	Full File System
FSAIoT	Forensic State Acquisition from Internet of Things
FTC	Federal Trade Commission
GDPR	General Data Protection Regulation
HAS	Home Automation System
IDFIF-IoT	Integrated Digital Forensic Investigation Framework
IDFIF-IoT	Integrated Digital Forensic Investigation Framework
IoT	Internet of Things
IVA	Intelligent Virtual Assistant
LEA	Law Enforcement Agency
LoS	Last-on-Scene
NBT	Next-Best-Thing
NVR	Network Video Recorder
OS	Operating System
PDF	Particle Deep Framework
PLIST	Property List
PRoFIT	Privacy-aware IoT-Forensics
SECSYS	Security System
SIIFF	Service-Interconnectivity-based IoT Forensics Framework
SWGDE	Scientific Working Group on Digital Evidence
UTC	Coordinated Universal Time
VR	Virtualized Resources
XVR	X Video Recorder

Chapter 1: Introduction

1.1 Introduction

The Internet of Things (IoT) represents a significant step forward in technology for humankind. IoT along with Artificial Intelligence (AI) are the emerging technologies that show promise for shaping the 21st century [1] [2], potentially influencing modern society in ways we may fail to conceive at the moment.

The term IoT, refers to a network of interconnected devices that can interact with each other, their surroundings, and the Internet, with or without human intervention. Future smart cities will be teeming with IoT devices like drones, smart vehicles, and more. These ubiquitous devices store, analyze, and share information, with their primary objective being to enhance the quality of the services they provide and improve the quality of human life.

Despite the many benefits that IoT offers, its widespread adoption introduces risks for its consumers [3] [4]. The low-security standards of IoT devices and privacy issues related to the data they process, are only two of the challenges that can pose threats to users' safety [5] [6] [7]. For example, the Federal Trade Commission (FTC) recently fined Amazon over $30 million for a series of privacy violations related to its Alexa assistant and Ring security cameras [8]. Furthermore, it is well-documented in the literature [9] [10] [11] that adversaries regularly exploit vulnerable IoT appliances to commit cybercrimes (either as a target or as a victim). On the other hand, IoT products frequently serve as “silent witnesses” to conventional crimes due to their omnipresence [12] [13]. In Greece, the latest case solved by the examination of such IoT devices was Caroline’s Crouch murder [14]. Thus, it is increasingly clear that these devices often store critical information that can be pivotal in various types of investigations.

Digital Forensics (DF) is the field of science that deals with investigating crimes that involve digital evidence. In particular, DF focuses on the legally acceptable extraction, analysis, and interpretation of data stored on, or transmitted to/from, digital devices. The devices in question are typically implicated in criminal activity, such as cybercrime or murder. To ensure the integrity, validity, and quality of DF results, many stringent policies and standard procedures have been established. However, the rapid proliferation and usage of interconnected devices, and their integral role in our daily lives, presents a new frontier for digital investigations.

This dissertation aims to address some of the identified challenges encountered when applying DF to the IoT realm, a process also referred to as *IoT forensics*. More specifically, it

seeks to fill a gap in the literature regarding the lack of knowledge about the artifacts that can be obtained from the forensic analysis of specific IoT appliances. Towards this goal, a plethora of IoT devices and the data they store were forensically evaluated. The results of this research led to important conclusions, the development of new open-source software, and updates to existing open-source digital forensic tools. The overall purpose of this dissertation is to support digital forensic examiners across the globe, as well as companies that develop digital forensic software, in their respective missions.

1.2 Problem Statement

Current statistics [15] about IoT-connected devices worldwide suggest their number will most likely reach 15 billion by the end of 2023, and could even end up as high as 29 billion by 2030. This vast number of devices greatly increases the chances of IoT appliances becoming “silent witnesses” to criminal activity. On the contrary, perpetrators may seek ways to tamper with this source of evidence to hide their tracks and avoid being arrested by Law Enforcement Agencies (LEAs).

Two IoT device categories that occupy a significant part of the IoT world are *Closed-Circuit Television* (CCTV) surveillance systems and *security systems* (SECSYS). Both of these product categories are widely accepted in smart homes, enterprises, etc.

According to research from Clarion Security Systems [16], there is one CCTV camera for every 10 people in London, totaling 945,562 cameras in the capital of the UK. Regardless of any legal and ethical implications that arise from such a large number of devices, footage from CCTV systems aids in crime reduction, both in terms of prevention and detection [17] [18].

On the other hand, SECSYS represent another category of pervasive IoT products. This category includes appliances like motion sensors, opening sensors, etc. which appear more and more often in modern smart homes, or elsewhere. These devices collectively form an IoT security system designed to protect the place where it has been installed along with its habitants from intruders and catastrophes, such as fire or water leakage. Despite the problems from the ongoing war, *Ajax Systems*, a Ukrainian-based manufacturer of such products, reported a 35% increase in revenue in its 2022 growth report [19], with its user base surpassing 1.8 million worldwide. This is indicative of the growing demand for these types of products.

After a thorough review of existing literature, a gap became evident, as highlighted in **Chapter 2** – **section 2.4**. Despite the omnipresence of IoT devices in the aforementioned

categories, as well as the potential for them to be linked with criminal activity, research regarding their digital investigation is limited. As a consequence, many such appliances that are produced by leading IoT manufacturers and thus are more likely to appear at crime scenes, either have not been forensically examined at all, or have only been partially analyzed. Hence, some of the critical information they store remains unexplored in both the literature and by digital forensic software. Thus, digital forensic examiners investigating a crime that involves such devices, not only may be unaware of these peculiarities but also may lack either the resources, the knowledge, or the time to study them. Furthermore, overreliance on the forensic software that they have at their disposal, may further risk overlooking evidentiary information that could potentially solve their case.

1.3 Research Objectives and Contribution

The aim of this dissertation is to address the above-mentioned problem by filling a part of the identified gap in the literature. In doing so, it seeks to assist digital forensic examiners around the world and companies that produce digital forensic software. The aforementioned goal can be deemed accomplished once the following research objectives are completed. These objectives mainly focus on the digital investigation of CCTV and SECSYS IoT devices and are detailed as follows:

i. Review the literature to identify CCTV and SECSYS IoT appliances manufactured by industry leaders that have either not been digitally investigated or have been partially examined but still contain unexplored data that are not addressed by major commercial forensic software.

ii. Collect the IoT devices identified in the previous objective, install them in controlled environments, and prepare them for digital investigation.

iii. Document and assess the findings accumulated from the digital investigation of the aforesaid IoT devices.

iv. Develop new open-source digital forensic software or contribute code to update existing ones as necessary. This would allow analysts to extract previously unexplored information related to the findings from the preceding forensic research.

v. Disseminate the results and insights from this research work to the broader DF community.

The main contribution of this research lies in the fulfillment of its research objectives. On one hand, **objectives ii**, **iii**, and **v** are achieved through the evaluation and presentation of the artifacts uncovered during the digital investigation of selected IoT devices. One point worth noting is that prior to the digital investigation of such devices, it is essential to ensure the process is lawful and respects individual rights. Throughout this dissertation, it is assumed that this matter has been addressed, and all the substantial and procedural legal requirements concerning the lawful collection and use of data/evidence are met, including authorizations for investigating these devices by the competent authorities. On the other hand, **objective iv** is accomplished via the development and update of open-source digital forensic software. The IoT devices investigated were chosen based on the findings of the literature review, as per **objective i**. The research contribution of this dissertation is outlined below:

i. Research, assess and display the artifacts derived from the digital investigation of *XIAOMI* SECSYS IoT devices.

ii. Study, review and exhibit the artifacts obtained from the digital investigation of both *DAHUA Technology* CCTV and SECSYS IoT appliances. Utilize the findings to enhance open-source digital forensic software to efficiently parse similar information in successive examinations.

iii. Explore, evaluate and demonstrate the artifacts derived from the digital investigation of *HIKVISION* CCTV systems. Exploit the results to develop new open-source digital forensic software as well as update existing ones to adequately interpret comparable information in the future.

iv. Examine, appraise and unveil the artifacts obtained from the digital investigation of *AJAX Systems* SECSYS IoT devices. Take advantage of the insights to create novel open-source digital forensic software to optimize the processing of analogous data in later inquiries.

Finally, the presentations and publications that were contributed to the literature as part of this doctoral dissertation are listed below:

Conference Proceedings (with jury system)

1	E. Dragonas, C. Lambrinoudakis, and M. Kotsis, “IoT forensics: Analysis of a HIKVISION’s mobile app,” in *DFRWS 2023 USA - Proceedings of the Twenty Third*	[20]

	Annual DFRWS Conference, Baltimore, Elsevier-Forensic Science International: Digital Investigation, Jul. 2023, p. 301560. doi: 10.1016/j.fsidi.2023.301560.	
2	E. Dragonas, C. Lambrinoudakis and M. Kotsis, "IoT Forensics: Investigating the Mobile App of Dahua Technology," *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, 2023, pp. 452-457, doi: 10.1109/CSR57506.2023.10224982.	[21]
3	E. Dragonas and C. Lambrinoudakis, "IoT Forensics: Analysis of Ajax Systems' mobile app for the end user," *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, 2023, pp. 446-451, doi: 10.1109/CSR57506.2023.10224992.	[22]

Journals

4	E. Dragonas, C. Lambrinoudakis and M. Kotsis, IoT forensics: Exploiting unexplored log records from the HIKVISION file system. *J Forensic Sci*. 2023; 00: 1–10. doi: 10.1111/1556-4029.15349	[23]
5	E. Dragonas, C. Lambrinoudakis and M. Kotsis, IoT forensics: Exploiting log records from the DAHUA technology CCTV systems. *J Forensic Sci*. 2023; 00: 1–14. doi: 10.1111/1556-4029.15401	[24]

Book Chapter

6	E. Dragonas, "IoT Forensics," in *The Hitchhiker's Guide to DFIR: Experiences From Beginners and Experts*, n.p., Leanpub.	[under publication]

Presentations in Conferences

7	E. Dragonas, "Forensic Analysis of Xiaomi IoT Ecosystem," *SANS Digital Forensics and Incident Response Summit, 2021*. Available: https://www.youtube.com/watch?v=4oVfHinPIz0&ab_channel=SANSDigitalForensicsandIncidentResponse	[25]
8	E. Dragonas, "Forensic Analysis of Xiaomi IoT Ecosystem - DFRWS USA 2021," *DFRWS USA, 2021*. Available: https://www.youtube.com/watch?v=zpCzctTUiWs&ab_channel=DFRWS	[26]
9	E. Dragonas, " IoT Forensics: Exploiting an unexplored piece of evidence in CCTV investigations " SANS Digital Forensics and Incident Response Summit, Tokio, 2023.	[27]

1.4 Dissertation Structure

The rest of this dissertation is organized as follows:

- **Chapter 2** provides an overview of the literature regarding digital forensics of CCTV systems, SECSYS, and other IoT devices while underlining the literature gap that was identified from its review.
- **Chapter 3** details the equipment that was employed, the methodology followed, and the results that were discovered during the digital investigation of *XIAOMI* SECSYS IoT products.
- **Chapter 4** demonstrates the equipment that was utilized, the methodology that was adopted, and the findings that were revealed during the forensic examination of both *DAHUA Technology* CCTV and SECSYS IoT appliances. Moreover, the code that was contributed to updating existing digital forensic open-source software is explained.
- **Chapter 5** displays the equipment that was used, the methodology that was followed, and the results that were uncovered during the forensic analysis of *HIKVISION* CCTV systems. In addition, the new digital forensic open-source software that was developed as part of this research is being introduced. Furthermore, the code that was written to enhance existing open-source software is demystified.
- **Chapter 6** outlines the equipment that was utilized, the methodology that was adopted, and the findings that were unearthed during the investigation of *AJAX Systems* SECSYS IoT appliances. Additionally, the digital forensic open-source software that was created as part of this work is being shown.
- The results of this dissertation are summarized in **Chapter 7**. Within this final chapter, the limitations of this research work are also highlighted and topics of future work are shared.

Chapter 2: Literature Review

2.1 Introduction

In this chapter of the dissertation, the literature relevant to IoT forensics and its challenges is presented.

In **section 2.2**, distinguishing research regarding the application of digital forensics to IoT appliances takes place. These studies are not only relevant to the CCTV and SECSYS IoT products but also related to the examination of other diverse IoT devices. Additionally, some DF frameworks and methodologies proposed for adoption in the IoT are included here as well. In **section 2.3**, some of the recognized challenges of IoT forensics are outlined. **Section 2.4** clarifies the research gap this dissertation attempts to fill as it arose during the literature review.

2.2 IoT Forensics

2.2.1 CCTV Systems

Current studies related to the digital investigation of CCTV systems heavily focus on the devices themselves. Studies have been conducted on how underlying file systems operate and how to efficiently recover video footage and metadata from them. In addition, algorithms and methodologies have been proposed on how to properly handle such systems from an investigation point of view. On the other hand, no published research was found related to the exploitation of the logging mechanism of a CCTV system or the companion applications that provide the ability to remotely operate such surveillance systems.

The way a surveillance system organizes, stores, and retrieves information relies on the file system it utilizes. The file system of such an IoT appliance can either be a standard one like EXT4 or a proprietary one developed by its manufacturer. Many companies, including *HIKVISION* and *DAHUA Technology*, ship their CCTV systems equipped with their proprietary file systems which they claim to provide faster indexing and video playback.

Li and Zuo [28] deciphered the proprietary file system of *DAHUA Technology*, namely the *DHFS* file system, and shared their insights on how to recover video footage from it. They focused on the *DHFS* structure and highlighted the main areas of the file system that could assist with this task.

Han et al. [29] and Sandeepa et al. [30] studied the proprietary file system of *HIKVISION*, namely the *HIKVISION* file system. They presented key areas within this file system and proposed methodologies for extracting saved video recordings.

Similarly, Gomm et al. [31] analyzed the *AVTECH* proprietary file system. This file system is utilized by the *GANZ* company. The authors were able to manually recover video records that even *GANZ* software could not. Their study denotes how fruitful can manual examination be when dealing with file systems that no digital forensic tool supports. Tobin et al. [32] also studied the same file system and proposed a reverse-engineering technique to retrieve and translate data from it, using an "*eavesdrop*" approach.

Gomm et al. [33] evaluated various approaches to CCTV forensics and introduced their methodology for acquiring and analyzing surveillance systems which they later applied in three case studies.

To stress how important is for CCTV manufacturers to integrate forensic readiness features into their products, Ariffin et al. [34] presented an anti-forensics framework for CCTV systems that allowed the permanent deletion of multimedia files stored on proprietary file systems.

Finally, Lu et al. [35] researched how to carve time metadata from video footage that gets recorded within proprietary file systems. To the same extent, Ariffin et al. [36] demonstrated a workflow that helped them recover video footage along with timestamps regardless of the underlying file system.

2.2.2 SECSYS

Within this section, insightful studies regarding the digital forensics of SECSYS IoT products are presented. Despite their quality, the number of these studies is rather small, indicating that more research is needed in this field.

Castelo Gómez et al. [37] extended the research of *XIAOMI* SECSYS IoT devices. In their work, they replicated some of the findings of this dissertation [25] [26]. Likewise, Giese [38] hacked many *XIAOMI* IoT devices to explore the information they store and provided insights related to their security.

Awasthi et al. [39] deeply examined the *SECURIFI Almond+* SECSYS and provided tips for acquisition as well as the interpretation of relative artifacts.

Mattia Epifani has systematically researched the application of DF to the IoT realm. In this instance [40], he demonstrated artifacts retrieved from the interrogation of IoT SECSYS appliances of the *Apple Homekit* ecosystem.

Kim et al. [41] investigated several IoT SECSYS devices such as *Samsung SmartThings* sensors. In their work, they combined artifacts recovered from multiple evidence sources to uncover all the available information they hid.

Servida and Casey [42] reviewed several IoT SECSYS devices (e.g., *iSmartalarm*) and evaluated the information they stored. In their work, the authors also managed to exploit vulnerabilities they discovered to gain access to these devices. They later disclosed these vulnerabilities to each manufacturer respectively.

Lastly, Hutchinson and Karabiyik [43] studied the *August* IoT SECSYS ecosystem and the artifacts that can be recovered from their Android and iOS companion applications.

2.2.3 Other IoT Devices

Apart from CCTV and SECSYS IoT devices, DF researchers have also studied other types of IoT products. The majority of these studies may have to do with the forensic exploration of Intelligent Virtual Assistants (IVA) but other IoT appliances have been examined as well. Their work is included in different sub-sections of this section.

2.2.3.1 IVA: Amazon Alexa

The most explored smart speakers in the literature are those powered by *Amazon's Alexa*. Orr and Sanchez [44] dug into an *Amazon Echo* to determine its forensic value. Chung et al. [45] assessed Alexa's ecosystem as well as introduced their *CIFT* toolkit which can be adopted in the forensic examination of such IoT products. Hyde and Moran inspected *Amazon Echo* and *Echo Dot* too and unraveled their capabilities [46]. Li et al. [47] analyzed a custom *Amazon Echo* (build on top of a Raspberry Pi) and used it for demonstrating their proposed IoT forensic model. Different generations of *Amazon Echo's* artifacts have also been investigated by Azhar and Bate [48], Pawlaszczyk et al. [49], Olufohunsi [50], Youn et al. [51], and more recently Lorenz et al. [52]. Lastly, a network forensic approach was adopted by Shin et al. [53] who explored the network traffic of an *Amazon Echo* (among other devices).

2.2.3.2 IVA: Google Assistant

On the *Google* counterpart, forensic scrutiny of *Google Home Mini* and its *Google Assistant* was the main topic of Moore's research [54]. Park and James [55] have examined *Google Home Mini* along with *Google Assistant*. *Google Home Mini* was part of Dorai et al. [56] study as well, while Akinbi and Berry [57] dived into the forensic interrogation of *Google Assistant*. On the other hand, Tristan et al. [58], Engelhardt [59], and Yildirim et al. [60] performed hybrid forensic examinations of both *Alexa* and *Google Assistant*. Last but not least, Barral et al. [61] tried to reverse-engineer the encrypted internal chip of *Google Home* speakers.

2.2.3.3 IVA: Apple Siri and Others

Apple's Homepod smart speaker powered by *Siri*, has been analyzed by Epifani [40] [62]. Jo et al. [63] have proposed several digital forensic practices for AI speakers' investigation, based on their findings from looking into several smart speakers like the *NAVER Clova*. A non-intrusive method for analyzing smart speakers' network traffic has been proposed by Lin et al. [64].

2.2.3.4 Other Appliances

As far as fitness trackers and smartwatches are concerned, Dawson and Akinbi [65] explored the artifacts from a *TomTom Spark 3* smartwatch. Kang et al. [66] investigated *XIAOMI Mi Band 2* and *Fitbit Alta HR* devices to discover their artifacts. Yoon and Karabiyik [67] examined *Fitbit Versa 2* and shared their findings. Hantke and Dewald [68] analyzed similar devices (*XIAOMI Mi Band 2, Fitbit Charge 2, and Huawei Band 2 Pro*) and developed an open-source tool to assist with their exploitation. Baggili et al. [69] evaluated forensic information retrieved from *Samsung Gear 2 Neo* and *LG G* smartwatches. Becirovic and Mrdovic [70] and Kehinde [71] uncovered the artifacts that can be obtained from the forensic analysis of a *Samsung Gear S3 Frontier* smartwatch. In this work, Kehinde also examined a *Fitbit Versa* smartwatch. The forensic analysis of a *Microsoft Band 2* was done by Quick and Choo [72].

A *Samsung Smart* TV was researched by Boztas et al. [73] for residual digital traces. Epifani [74] [75] manually went through forensic images from an *LG* TV and an *Ematic TV OS Box,* and an Amazon Fire TV Stick was investigated by Hadgkiss et al. [76].

Zhou et al. [77] and Epifani [78] demystified evidentiary information that can be recovered from intelligent robot vacuum systems. Epifani additionally examined an image from a *Samsung Refrigerator* [79].

The *Ring* video doorbell was interrogated by Winkelman et al. [80]. Diverse IoT devices, ranging from a *TP-link Archer C1900* to a *Bitdefender Box 2*, were forensically analyzed by Hutchinson et al. [81].

Lastly, one illustrative instance among the many studies on smart vehicle forensics is the research conducted by Le-Khac et al. [82]. They addressed the challenges of this field as well as analyzed the mobile data traffic from an *Audi*, a *VW*, and a BMW car.

It is becoming obvious there are so many IoT devices out there, that the need for IoT forensics research is great and pretty much, ever-growing. Regardless, studies like the aforementioned ones are crucial for the digital forensics and incident response (DFIR) community and can help the reader of this dissertation to get a grasp of where the current literature concerning DF of IoT devices lies.

Except for the digital investigation of IoT devices, promising research has also been made in terms of new frameworks and models suitable for IoT forensics. Some of them are collected in the following sub-section.

2.2.4 Frameworks/Models

2.2.4.1 Frameworks

Meffert et al. [83] suggested the adoption of a general framework they termed *Forensic State Acquisition from Internet of Things* (*FSAIoT*). This framework aimed to tackle the challenge of the volatile information that an IoT device stores by utilizing a controller which would save the state change of each registered IoT appliance along with the date/time information of each change event.

Goudbeek et al. [84] envisioned a forensic investigation framework for smart home automation systems (HAS) that comprised seven phases. Not all phases were needed for real investigations. This framework should help examiners identify, collect, and analyze evidentiary data related to HAS.

Kebande and Ray [85] initially proposed their generic *Digital Forensic Investigation Framework for IoT* (*DFIF-IoT*). This framework had the benefit to comply with the *ISO/IEC 27043: 2015* international standard for information technology. A couple of years later, Kebande et al. [86] enhanced and extended this framework with another one, namely the *Integrated Digital Forensic Investigation Framework* (*IDFIF-IoT*) for an IoT ecosystem.

Babun et al. [87] developed the *IoTDots* framework. This novel framework was capable of both detecting forensic-relevant events from IoT devices and companion applications and enabling the storage of this information via integrating a logging mechanism. Interestingly enough, all the aforementioned frameworks were comprehensively evaluated by Hassan et al. [88].

Koroniotis et al. [89] introduced their network forensics framework, namely the *Particle Deep Framework* (PDF), which can be used for identifying and tracing attack behaviors in IoT networks.

Shakeel et al. [90] unveiled a *Blockchain-Assisted Shared Audit Framework* (*BSAF*) that can assist with the analysis of digital forensic data in the IoT platform. Their proposed framework was designed to trace the source of data scavenging attacks within *Virtualized Resources* (VR).

Surange and Khatri [91] studied the heterogeneity of the information IoT devices store and shared their conceptual framework, which would facilitate the acquisition and analysis of evidentiary data, using a unified repository of information collected from an IoT ecosystem.

Recently, an interesting framework was introduced by Jacob and Nisbet [92]. The authors provided a novel concept of exploiting radio frequency signals to discover and locate the wireless sensing appliances of a digital crime scene.

2.2.4.2 Models

Oriwoh et al. [93] envisioned a zone-based forensic method for approaching IoT-related investigations, namely the *1-2-3 Zones*. Using these zones as a guide, the investigator should be able to identify where each piece of evidence is located and then prioritize their collection. In the same work, the authors proposed the *Next-Best-Thing* (*NBT*) triage forensic model to be used in conjunction with the *1-2-3 Zones*. Their model focused on collecting residual evidence from the crime scene.

Zawoad and Hasan [94] conceptualized a trusted repository where all evidence related to IoT devices would get stored. The implementation of this concept, which would result in easier collection and preservation of evidence, was labeled as the *Forensics-aware IoT* (*FAIoT*) model.

The enhanced *IoT-based Digital Forensic* model propounded by Perumal et al. [95], integrated many forensic models that were previously recommended. Another improved model that combines others, is the *Last-on-Scene* (*LoS*) forensic model Harbawi and Varol [96] discussed.

Qatawneh et al. [97] presented their *Digital Forensics Investigation Model* for IoT (*DFIM*). The proposed *DFIM* consisted of seven stages and considered principles such as security, privacy accuracy, performance, etc.

Nieto et al. [98] formulated the *Privacy-aware IoT-Forensics* (*PRoFIT*) model. Their model reflects on privacy principles and implements them to protect citizens' data stored within their IoT devices. This way they aim to stimulate the cooperation between civilians and LEAs.

Scheidt and Adda [99] described their theoretical model of a *Hybrid Forensic IoT Server*, where the unique "*DNA*" characteristics (e.g., serial number/location, etc.) of each IoT device are registered on the server. This would help identify devices and exchange information related to these IoT devices across similar servers installed around the world.

The IoT-focused modular digital forensic model put forward by Hilgenberg et al. [100] was created as an easy-to-follow and implement methodology in any ongoing investigation. In their work, the authors applied their model to a couple of use cases to highlight its advantages over other models.

Agbedanu and Jurcut [101] envisioned the *Blockchain-Based Forensic Model* for IoT (*BLOF*), a model which utilizes blockchain to prevent the admissibility of tampered logs as evidence, in criminal investigations.

More recently, Kim et al. [102] introduced their *Service-Interconnectivity-based IoT Forensics Framework (SIIFF)*. Their model is partially based on previous ones and can be used to reveal the relationships between interconnected IoT things and better understand the target IoT environment. Another recent study conducted by Akinbi et al. [103] focused on the latest blockchain-based IoT forensic investigation process models. The authors reviewed them to evaluate their efficiency in real investigations.

Finally, the investigator should also be aware of the best practices and notes on IoT forensics that the *Scientific Working Group on Digital Evidence* (*SWGDE*) published [104] [105]. These resources may come in handy when performing an on-site seizure of IoT devices.

The reviewed literature, presented in this section has shed light on various characteristics of IoT forensics, offering valuable insights into the current state of knowledge. However, it also became apparent that several challenges are associated with this field. These obstacles encompass both legal and technical aspects, demanding attention and further exploration.

Therefore, in the subsequent section, many significant challenges related to IoT forensics are enumerated. By acknowledging these challenges, the way can be paved for advancements and innovative solutions to them.

2.3 Challenges related to IoT forensics

More than 20 studies were discovered that either directly pinpoint complications related to IoT forensics or indirectly mention some of them. The classification of challenges varied significantly among these works. Therefore, the issues most frequently encountered in these researches have been categorized and are listed below.

2.3.1 Forensic Challenges

The difficulties summarized within this section are connected with the digital investigation of IoT devices. These challenges have been grouped based on the main phases of a forensic analysis, namely *Identification*, *Collection*/*Preservation*, *Examination/Analysis*, and *Presentation/ Reporting*.

2.3.1.1 Identification

When it comes to obstacles related to the identification phase, the following are among those that have been recognized in the literature [3] [4] [6] [7] [93] [94] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118]:

- **Data Location**: It is really hard to determine the location of evidentiary data within an IoT crime scene. Crucial information could be residing in multiple local locations (companion mobile app, IoT device, etc.), remote places (e.g., server in other country), or a combination of those. This can be daunting for the examiner on site.
- **Device Type**: Another equally hard task for the investigator is to evaluate all the diverse IoT appliances that could be important and could end up in a crime scene, ranging from smart refrigerators to smart vehicles or else. These devices could be extremely difficult to be secured, seized, and transported back to the lab for further analysis.
- **Device and Data Proliferation**: The proliferation of interconnected devices and the amount of forensic data they store, is another problem widely discussed. Having to decide which IoT appliances are relevant to the case and need to be collected and which ones to discard can be a time-consuming assignment.

2.3.1.2 Collection/Preservation

It is very challenging for an investigator to collect and preserve evidence from an IoT crime scene. Some of the major challenges concerning this concept, have been studied in the literature and are summarized below [3] [4] [5] [7] [42] [94] [106] [110] [111] [112] [114] [115] [116] [119]:

- **IoT Device Characteristics**: The IoT devices have typically limited storage, finite power, lightweight CPU, and restricted network communication capabilities. These characteristics vary based on the duties each appliance is designed to perform, how regularly it interacts with its environment, etc. For instance, a motion sensor that solely detects movement and reports it to its smart hub should have fewer complex features than the hub which has to collect this information from all deployed sensors, process it, and push it to the cloud. Their variance inevitably affects both their collection and preservation.

- **Lifespan of Evidentiary Data:** Due to the restriction of storage in IoT devices, the lifespan of information stored within them is rather short and data can be easily overwritten. Particularly, since IoT appliances have limited storage, most of the data they generate gets eventually transferred and stored in the cloud. Therefore, the cloud becomes one of the main sources of evidence in IoT forensics. Consequently, certain implications related to the collection of evidence from the cloud also apply here (e.g., lack of physical access to the cloud servers).

- **Crime Scene Contamination**: Another point of consideration that derives from the low storage capabilities of IoT devices is that any events triggered by the first responders on the scene could potentially fill the appliance memory, prompting it to overwrite the older relevant events. These events would most likely get pushed to the cloud, leaving the memory of the device contaminated with the events created by the intervention at the crime scene. In cases where the collection of cloud evidence is prohibited, this action could jeopardize the investigation.

- **Securing the Chain of Custody**: Ensuring the chain of custody is challenging in IoT forensics. It is almost impossible not to contaminate the crime scene during the collection of evidence. Even if this action is necessary, it needs to be justified in court so as the collected evidence to be accepted as legitimate. Also, it is hard to choose which handling procedures to follow in such a scenario. Furthermore, the integrity of collected evidence from IoT devices is difficult to preserve, due to both their dynamic nature and the lack of forensic tools that can prevent any accidental modifications to them.

- **Data Storage Period in the Cloud**: Since most of the evidentiary data is stored in the cloud, collecting this source of evidence is paramount for the investigation. However, the storage period for this information is determined by each cloud service provider (CSP) as well as the legislation of the countries where the CSP operates. The fact that this period varies between 6 months and up to 1 year in most cases, can severely impact the retrieval of this evidence source.

- **Lack of Knowledge and Training**: Most investigators do not know how to properly handle an IoT crime scene. They are not trained properly to tackle such a dynamic environment. For example, should the first responders shut down the IoT device before collection or should they try to acquire it while it's operating? The knowledge to answer such questions is gained only through special training which would instruct responding investigators on how to properly seize and secure each type of IoT appliance.

2.3.1.3 Examination/Analysis

The aim of the examination/analysis phase is to extract meaningful information from collected evidence. Some of the most important implications related to this phase are outlined below [6] [7] [94] [106] [110] [111] [112] [115] [116] [117]:

- **Data Format**: IoT devices store their information in diverse file types such as databases, logs, etc. Additionally, data regarding a particular IoT appliance can be obtained from various evidence sources (mobile application, internal storage, cloud servers, etc.). This heterogeneity requires strenuous effort for the thorough examination of all available evidence.

- **Lack of Standardization**: The multitude of log records (user events, network logs, application logs, etc.) that get created by IoT devices do not meet specific standards. Instead, manufacturers record only the information of their interest, in their desired format, and even by utilizing their own proprietary operating/file systems. Due to the lack of standardization and uniformity, analyzing these records is a demanding assignment.

- **Limitations in the Currently Available Forensic Tools/Lack of IoT Forensic Tools**: The available forensic tools encounter substantial challenges in effectively collecting and parsing evidence related to IoT devices. These limitations can be either ascribed to the rapid and ongoing evolution of IoT products, which outpaces their capabilities or the fact that these tools were not developed for the examination of IoT appliances in the first place. The

complexity and diversity of IoT systems, coupled with their extensive range and volume of data, create unique hurdles for current forensic software and underscores the need for developing new tools, specifically for such infrastructures.

- **Lack of Knowledge and Training**: This challenge also apply in this phase. Examiners need training on how to analyze the collected evidence. Research of IoT devices which could be used as a reference/starting point from investigators is both required and highly encouraged.

2.3.1.4 Presentation/Reporting

This phase involves preparing and delivering the results of the forensic investigation in a manner that is comprehensible and legally defensible. Clearly, presenting findings in court from digital investigations related to IoT forensics poses distinct challenges, some of which have been identified and are listed here [3] [111]:

- **Complexity of IoT Ecosystems**: IoT appliances often operate within complex networks comprising of multiple different devices, manufacturers, and communication protocols. Presenting information regarding these complexities in a way that is intelligible to non-experts can be daunting.

- **Interpreting Data Correctly**: Data from IoT devices may come in different formats and be based on different operating/file systems depending on the manufacturer and type of device. Incorrect interpretation of such data can lead to misleading conclusions.

- **Technical Expertise**: Conveying technical findings to a lay audience (e.g., jury or judge) requires both thorough technical and legal knowledge as well as a capacity to communicate complex technicalities related to IoT technology in an accessible and understandable manner. This can be a significant challenge, especially in this rapidly evolving field.

2.3.2 Security Challenges

The expansion of the IoT has brought about a wide range of security challenges that can impact the field of IoT forensics. Here are some of the key challenges that were met during literature review [3] [5] [6] [106] [110] [113] [116] [117] [120]:

- **Lightweight Security Measures**: IoT appliances have limited CPU, storage, and power resources. Hence, robust security measures cannot be installed on them. Furthermore, many manufacturers do not patch the vulnerabilities of their IoT products regularly or at all.

Therefore, these devices are often susceptible to hacking and data manipulation. This can jeopardize the integrity of the digital evidence obtained from them.

- **Lack of Standardization**: IoT lacks standardization in terms of device design, data formats, and network protocols, which can exacerbate security vulnerabilities and hamper their digital investigation.
- **Attack Surface**: IoT provides a wide security attack surface due to the constant introduction of new and diverse devices, equipped with various operating systems (OS), and supporting multiple communication protocols. Potential security breaches could involve DDoS attacks conducted via compromised IoT appliances, unauthorized access to CCTV and IP cameras, and more.

2.3.3 Legal Challenges

The legal challenges that were found during the review of existing literature [3] [5] [6] [7] [93] [108] [110] [115] [116] [118] [120] [121], highlight the need for updated legislation in order to cope with the ever-evolving IoT technology. Some of the most common implications discussed are demonstrated here:

- **Data Privacy**: In the highly dynamic IoT environment, multiple interconnected devices often collect personal and sensitive information, without user awareness and consent, raising serious concerns about user privacy. For instance, IoT products like fitness trackers track sensitive data including users' steps, geolocation, health status, or even medical records which are later pushed to the manufacturer's cloud. Legislation such as the General Data Protection Regulation (GDPR) includes legal requirements dictating that the processing of such personal data should be carried out in a manner that is lawful, fair, and transparent. However, applying data protection regulations to this type of information presents a significant challenge due to the rapid evolution and expansion of the IoT landscape. Conducting forensic investigations within this context further complicates matters, as investigators must carefully navigate legal intricacies to avoid infringing upon privacy laws.
- **Multijurisdictional Issues**: IoT devices store and transfer their data employing numerous cloud servers across different geographical locations and legal jurisdictions. Determining the applicable legal framework for each scenario and ensuring lawful access to the data can be complicated and sometimes even impossible. For example, when dealing with an incident involving IoT appliances, one of the primary challenges lies in establishing the appropriate

jurisdiction for prosecution. Decisions need to be made about whether the case falls under the jurisdiction related to where the data is stored, the location of the IoT device under investigation, or the perpetrator's location.

- **Admissibility of Evidence**: Ensuring that digital evidence acquired from IoT devices is admissible in court is another notable legal implication. In the realm of IoT, proving that the evidence was collected and analyzed using reliable forensic methods can be really difficult. This is largely due to the obstacles discussed in the preceding sections. Similarly, ensuring and demonstrating that the chain of custody was properly maintained to preserve the integrity of the evidence also presents significant challenges.

2.4 Research Gap in the IoT forensics

After the extensive review of existing literature in the field of IoT forensics, several potential areas of exploration emerge. One such area stems from the narrow knowledge of the forensic artifacts that can be obtained from the digital investigation of SECSYS and CCTV IoT products (See **sections 2.2.1** and **2.2.2**). Moreover, the currently available forensic tools face challenges in efficiently parsing evidence collected from these devices.

Therefore, based on the findings from the literature review, the following unexplored evidence sources were chosen to be examined as part of this dissertation:

- ***XIAOMI* SECSYS IoT Devices**: *XIAOMI* is considered one of the biggest IoT manufacturers worldwide [122]. The company offers its customers a mobile application to remotely control SECSYS and other IoT devices within its ecosystem. This mobile application has not been previously explored in existing literature or by digital forensic software, making it a selected subject for exploration in this dissertation. The objectives of its research are detailed in **contribution i** (See **Chapter 1** – **section 1.3**) and its results are presented in **Chapter 3**.
- ***DAHUA Technology* CCTV and SECSYS IoT Devices**: According to third-party research referenced within its webpage [126], *DAHUA Technology* has been estimated to be the second-largest supplier of video surveillance equipment in the world since 2014. The company provides CCTV systems equipped with file systems that store various log records. It also offers its customers a mobile application to remotely control its SECSYS IoT devices. Neither the log records stored by its CCTV systems nor its mobile application have been previously studied, making them selected subjects for exploration in this dissertation. The

goals of their research can be found in **contribution ii** (See **Chapter 1** – **section 1.3**). The findings are then presented in **Chapter 4**.

- ***HIKVISION* CCTV IoT Devices**: Based on a recent analysis from *Research and Markets* [123], *HIKVISION* is considered among the global surveillance camera market leaders. The company offers CCTV systems that utilize a proprietary file system for storing a variety of log records. Additionally, they provide a mobile application that allows customers to remotely operate these IoT devices. No prior research has focused on the log records from its CCTV systems or this mobile application, making them chosen subjects of exploration for this dissertation. The objectives of their research are outlined in **contribution iii** (See **Chapter 1** – **section 1.3**). The results are then presented in **Chapter 5**.
- ***AJAX Systems* SECSYS IoT Devices**: According to its 2022 growth report [19], *AJAX Systems* operates in *169* countries, has seen a *35%* increase in its revenue, and its end users surpassed *1,8* million worldwide. The company provides customers with a mobile application that allows for remote control of SECSYS and other IoT devices in its ecosystem. This application hasn't been examined in prior literature or through digital forensic tools, prompting its selection as a topic of interest for this dissertation. The aims of its research can be found under **contribution iv** (See **Chapter 1** – **section 1.3**) and its findings are presented in **Chapter 6**.

This dissertation aims to partly address the identified literature gap by conducting a thorough forensic analysis on the aforementioned evidence sources. Its focus is on gathering, evaluating and sharing artifacts identified during these examinations. It also discusses the development of new digital forensic software and updates to existing ones utilizing the obtained artifacts. Consequently, this dissertation will form a valuable resource not only for investigators in this field but also for companies that develop digital forensic software.

In the subsequent chapters, the research contribution of this dissertation to IoT forensics is presented and discussed in detail.

Chapter 3: Digital Investigation of *XIAOMI* SECSYS IoT Devices

3.1 Introduction

In this chapter, the forensic analysis of *XIAOMI* IoT appliances is presented. While these devices predominantly belong to the SECSYS category of products, other smart home appliances have also been incorporated into this research work. *XIAOMI* is considered one of the biggest IoT manufacturers worldwide [122]. *XIAOMI* along with its subsidiaries and third-party partners provides a huge variety of IoT products including security devices, smart bulbs, power sockets, etc. As a consequence, IoT appliances produced by *XIAOMI* are becoming more and more likely to appear in smart home investigations.

3.1.1 Contribution of this chapter

This chapter addresses the objectives outlined in **contribution i** (See **Chapter 1** – **section 1.3**) by forensically examining a subset of *XIAOMI* IoT devices. Significant highlights from this research are as follows:

- The presentation of the way the *XIAOMI* mobile application for the end-user operates.
- The presentation of the artifacts that can be obtained from the forensic analysis of its mobile application and how they could be used in real cases.

3.1.2 How this chapter is organized

The subsequent sections of this chapter are structured in the following manner: In **section 3.2**, the equipment used for this research work is listed. Similarly, in **section 3.3** the DF methodology adopted is displayed. The results of this assessment are demonstrated in **section 3.4**. Discussion about the findings takes place in **section 3.5** while **section 3.6** concludes this chapter.

3.2 Equipment

A *XIAOMI Redmi Note 6 Pro* mobile device with Android 9 was employed for this research work. The device was rooted using *Magisk* [124] in order to gain full file system access to it. A plethora of *XIAOMI* IoT products were researched, including sensors (Motion, Door and Window, etc.), a wireless Switch, a Control Hub, Robot Vacuum, and more. The complete list of devices of this research work is shown in Figure 1. These devices interacted with each other, as well as with the

mobile application and the *XIAOMI* cloud, by using the ZigBee, the Bluetooth and the WiFi wireless network protocols. Their communication structure is illustrated in Figure 2.

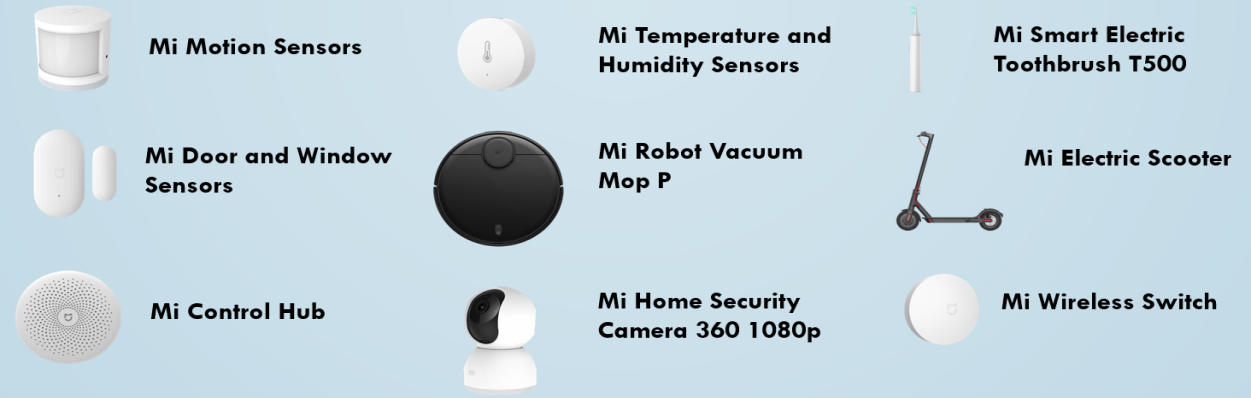

Figure 1 List of IoT devices examined in the XIAOMI research work

XIAOMI offers a dedicated mobile application, known as the *Mi Home* app (with its Android package name being *com.xiaomi.smarthome*), to allow its consumers to control their IoT appliances. Specifically, users of *XIAOMI* IoT products can utilize this app to install, set up, and monitor their devices. The *6.7.700* version of this application was analyzed in this research work.

The forensic analysis of this work was performed on a Windows 10 Pro workstation. Android *SDK Platform Tools (including ADB)* [125] was used for retrieving evidentiary data from the Android phone. For the examination of the collected evidence, *X-Ways Forensics* [126] was utilized. In addition, since the application partly stored data in SQLite databases and XML files, *DB Browser for SQLite* [127] and *Notepad++* [128] were also used for viewing them. The hardware equipment of this work is displayed in Table 1 whereas software used is shown in Table 2.

Table 1 Hardware equipment used in the XIAOMI research work

Hardware	Model/Version
XIAOMI Mi Motion Sensor (2 pcs)	RTCGQ01LM
XIAOMI Mi Door and Window Sensor (2 pcs)	MCCGQ01LM
XIAOMI Mi Temprature and Humidity Sensor (2 pcs)	YTC4042GL
XIAOMI Mi Control Hub	DGNWG05LM
XIAOMI Mi Wireless Switch	WXKG01LM
XIAOMI Mi Smart Electric Toothbrush T500	MES601
XIAOMI Mi Robot Vacuum Mop P	STYTJ02YM
XIAOMI Mi Home Security 360º Wireless camera with SanDisk 32GB microSD	MJSXJ05CM SanDisk Ultra
XIAOMI Mi Electric Scooter	BHR5389GL
XIAOMI Redmi Note 6 Pro	M1806E7TG

PC workstation with: ▪ Windows 10 ▪ Intel i7	 1809 9700K

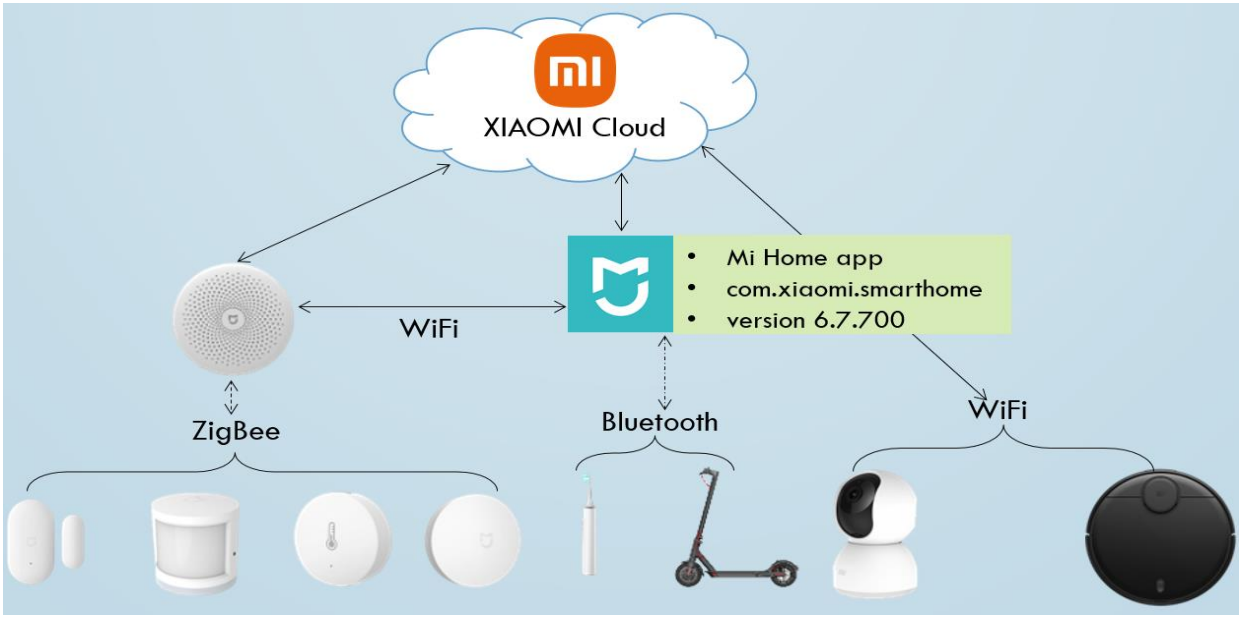

Figure 2 Communication within XIAOMI IoT ecosystem

Table 2 Software used in the XIAOMI research work

Software	Version
Android SDK Platform tools for Windows	31.0.1
Magisk	23.0
DB Browser for SQLite	3.12.2
Notepad++	8
X-Ways Forensics	20.4

3.3 Methodology

To begin with, the *Mi Home* application was installed on the Android mobile device. Afterwards, the IoT products of this research work were placed around so as to form a smart home, as illustrated in Figure 3. Next, two new *XIAOMI* accounts were created as part of this research work. One account was utilized for interacting with the application, while the other was employed to share access to the smart home. The mobile application was used to configure the installed IoT appliances and bind them with the first *XIAOMI* account. The IoT ecosystem that was set up was used for a period of 2 months. Within this period, different device configurations were chosen from time to time, the events recorded by the devices were viewed, and diverse automation scenarios

were chosen for some of the devices in order to trigger the creation of more artifacts. For all of these actions, the mobile application was utilized.

After the period of 2 months, the evidence collection took place. Unfortunately, due to lack of specific equipment and required knowledge (e.g., chip off methodology), evidence from the internal storage of the employed IoT devices was not acquired. Similarly, since there was no established method to retrieve evidentiary data pushed to the *XIAOMI* Cloud by the IoT products, this type of data also remained irretrievable. On the other hand, information residing within the mobile application could be collected and hence was the type of evidence this work focused on.

Figure 3 The IoT smart home of the research work

In order to collect this source of evidence, *ADB* from the *Android SDK Platform Tools* was deployed. Using this tool, the mobile application's data located at the *"/data/data /com.xiaomi.smarthome*" directory was copied to the workstation. This process was repeated several times over the two-month period, each repetition following a series of experiments conducted with the app. This method was employed to monitor any variations among the discovered artifacts. After collecting the evidence from the aforementioned source, its examination (using the software described earlier) followed. The findings obtained from the analysis of the *Mi Home* smartphone application are presented in the following section.

3.4 Results

The results of this research are detailed separately, based on the evidentiary information.

3.4.1 Artifacts related to the application

The "*/shared_prefs/one_track_pref.xml*" file recorded the timestamp when the application was first launched (in *UNIX milliseconds* format). It also stored the latest installed version of this app.

3.4.2 Artifacts related to the smart home

The "*/files/sh_home_id*" and "*/shared_prefs/home_room_transfer_state.xml*" files held the unique ID that was assigned to the smart home that was created as part of this work. This ID was labeled "*homeid*" and can be used to refer to this specific smart home across the vast *XIAOMI* IoT universe. However, the most crucial detail retained in these files was the location (country) of the *XIAOMI* cloud server, which is where the data associated with this smart home was pushed. In this research work, this server was found to be located in Germany/Deutschland (country code "*de*").

The "*/shared_prefs/home_roomv_manager_sp_.xml*" file was another important artifact with regard to the smart home under investigation. This file contained the smart home ID, the name its user gave it, its geolocation (if location permission was granted), and information related to the rooms of the smart home (names, unique room ID, etc.). Part of the content of this file is shown in Figure 4.

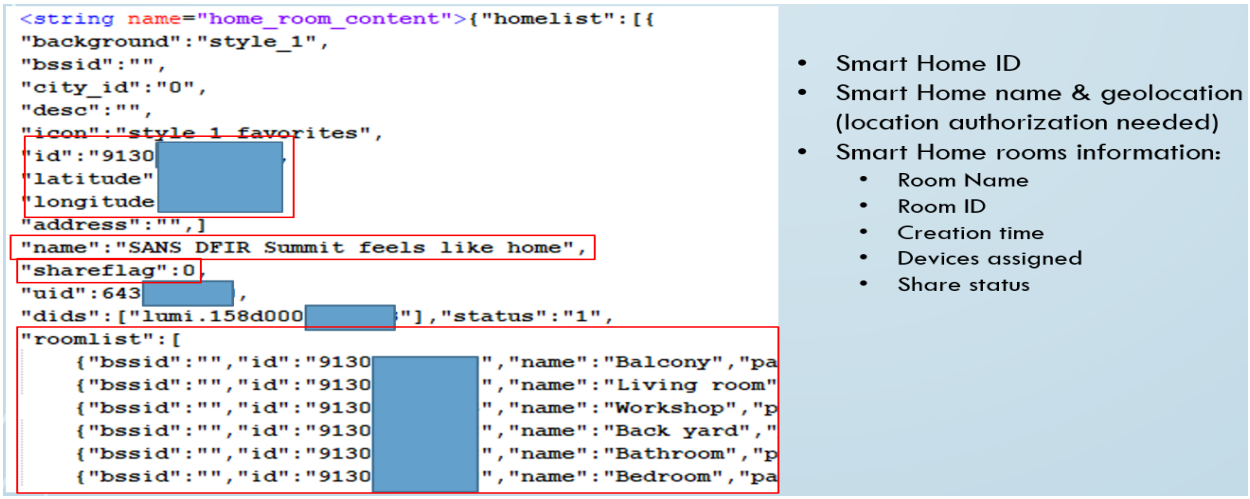

Figure 4 Part of the content of the "home_roomv_manager_sp_.xml" file

As far as the automation schedule of the smart home is concerned, the "*/shared_prefs/MD5(UserID)*[1]*scene_list_cache.xml*" is the file that the investigator should look for. This file contained information about the configured automation scenarios (a.k.a. "*smart scenes*"), including the devices they involved, when they were created, etc. Conversely, within this file no historical log records of when each smart scene was triggered were found. An example of the data kept in this file is demonstrated in Figure 5.

3.4.3 Artifacts related to the XIAOMI accounts

Within the "*shareuserecord*" table of "*/databases/miio.db*" SQLite database, useful information about both the *XIAOMI* accounts was uncovered. This included the users' email, nickname, phone, unique user ID, and the timestamp when sharing access to the smart home was enabled. This artifact can be pivotal when the investigator wants to determine who the registered owner of the smart home was as well as who else had access to it.

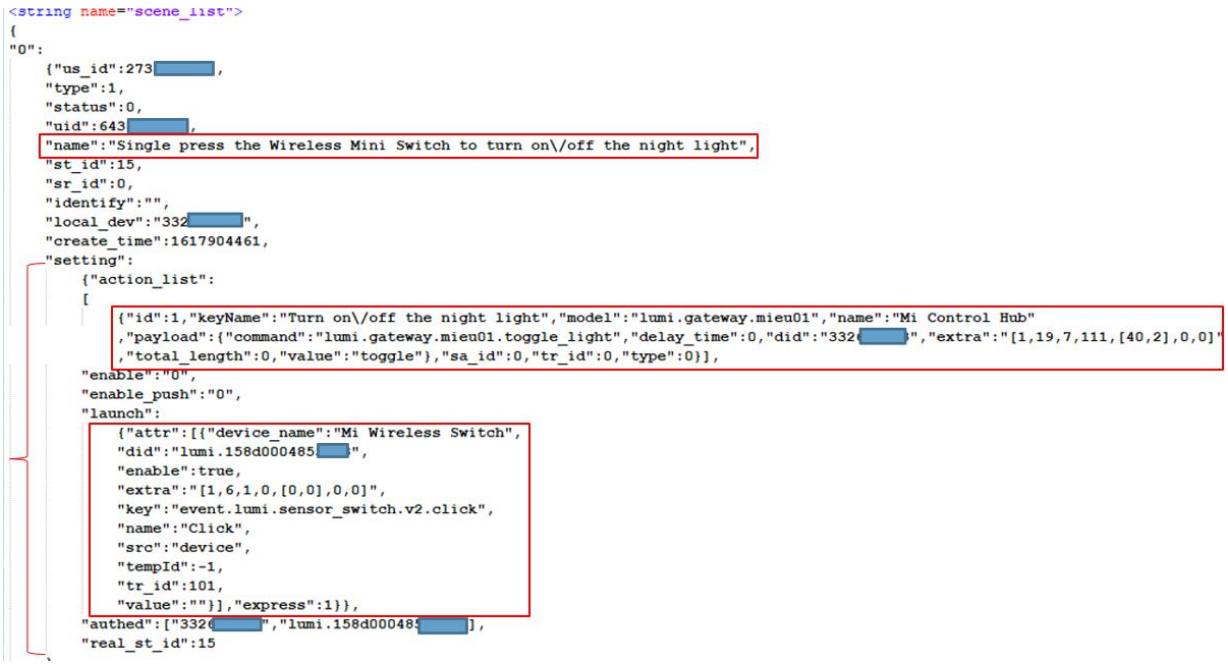

Figure 5 Part of the content of the "MD5(UserID)scene_list_cache.xml" file

Another file that could be found interesting even if it was not always populated, was the "*/shared_prefs/shared_user_info_list_UID.xml*". This XML file contained evidence about the logged in XIAOMI account, such as when the account was created (in *UNIX milliseconds* format).

[1] The term *'MD5(UserID)*' indicates that the filename begins with a MD5 hash value, which corresponds to the MD5 hash value of the unique ID (labeled '*uid*') of the XIAOMI account that has been logged into the application.

3.4.4 Artifacts related to the IoT devices

3.4.4.1 Mi Motion sensors, Mi Door and Window sensors, and Mi Wireless Switch

Under the "*/files/plugin/install/rn/*" directory several folders beginning with "*100****" were discovered. Every folder represented a different IoT device type. Inside these folders, there were several subfolders and files, each containing information related to their respective IoT product.

The key file for these appliances' digital investigation was the "*config.xml*" which was positioned in the *"/files/plugin/install/rn/100****/10*****/data"* directory of each type of device. This XML file logged the *20* latest events the device "*sensed*" along with some metadata (e.g., the timestamp of the event occurrence). For instance, some of the latest records stored within the "*config.xml*" file of one of the *Mi Motion* sensors is presented in Figure 6.

3.4.4.2 Mi Humidity & Temperature sensors

The "*config.xml*" file for the *Mi Humidity & Temperature* sensors is structured similarly to the previously mentioned files, with one exception. The amount of records this file can store is based on the application's cached data and it is not fixed (e.g., *20* latest records). The more data the user retrieves from the *XIAOMI* cloud for this type of IoT appliance (via viewing its measurements), the more records would be found in this file.

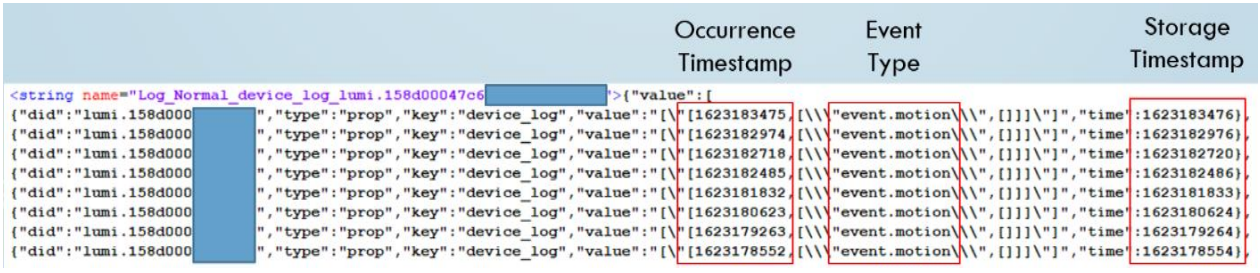

Figure 6 Part of the content of a "config.xml" file

3.4.4.3 Mi Control Hub

The *Mi Control Hub* is the IoT smart hub where all the aforementioned appliances get connected to. These devices communicated with the hub via the ZigBee protocol. This device is often considered the "*heart*" of a smart home, as it can orchestrate the rest of the aforesaid appliances, it is the only device of them capable of pushing events to the cloud, and without it, any events recorded by the endpoints would never reach the end user.

The "*config.xml*" file for the *Mi Control Hub* maintained specific information about this IoT device. This entailed the device's "Guard status"[2], its "Guard settings" (Alert Conditions, etc.), the IoT products currently paired with the *Mi Control Hub*, and the latest historical alert records.

3.4.4.4 Mi Robot Vacuum Mop P

No "*config.xml*" file was found for this device. On the contrary, the "*catalystLocalStorage*" table of the "*/databases/RKStorage*" SQLite database contained data related to this product. More specifically, this table kept metadata about the latest cleaning event, a base64-encoded image of the map created when the device first scanned the smart home, and more. The smart home map was stored in the "*value*" column. It was decoded and is presented in Figure 7.

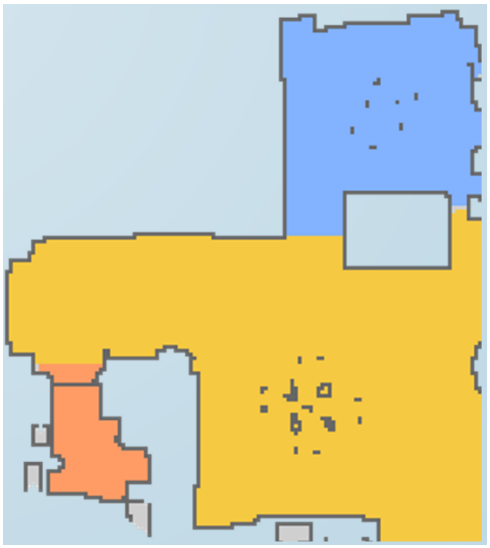

Figure 7 The map the robot vacuum created and stored within the "RKStorage" database

3.4.4.5 Mi Home Security 360° camera

No "*config.xml*" file was found for this appliance either. If a microSD card is utilized, then all the camera's available recordings would be put there. If the users have chosen to create and store image/video recordings to their mobile phone while viewing the camera's footage, then these media files can be found under "*/DCIM*" directory in a folder named after the unique ID of the IP camera. *XIAOMI* also offered a feature labeled as "*surveillance assistant*". This feature created specific clips from the recordings of the device that were related to security (e.g., when the camera

[2] Guard status: The *Mi Control Hub* can be configured to act as an alert siren when any of its connected sensors (e.g, Motion sensor, Door and Window sensor) record an event (motion detected, etc.). If "Guard Status" is enabled, then this mode is on.

detected movement or/and a person). These clips were pushed to *XIAOMI* cloud for 7 days, in which period the users can view these events whenever they wanted. When this feature was enabled in this research work, records related to these clips were found within the "*/databases/mijia_camera_cloud_video.db*" SQLite database.

3.4.4.6 Mi Smart Electric Toothbrush T500 and Mi Electric Scooter

Unfortunately, no useful artifacts were found for the Bluetooth devices in the application's space. Therefore, these appliances were examined live through the application. Bluetooth was enabled and the application was used to monitor information about the aforementioned products. Needless to say, this is not a forensically sound method to retrieve/examine evidence. Using this process, the examiners can retrieve records when the smart toothbrush was used. Figure 8 provides an example of these records. They can also retrieve some basic information about the electric scooter (e.g., serial number).

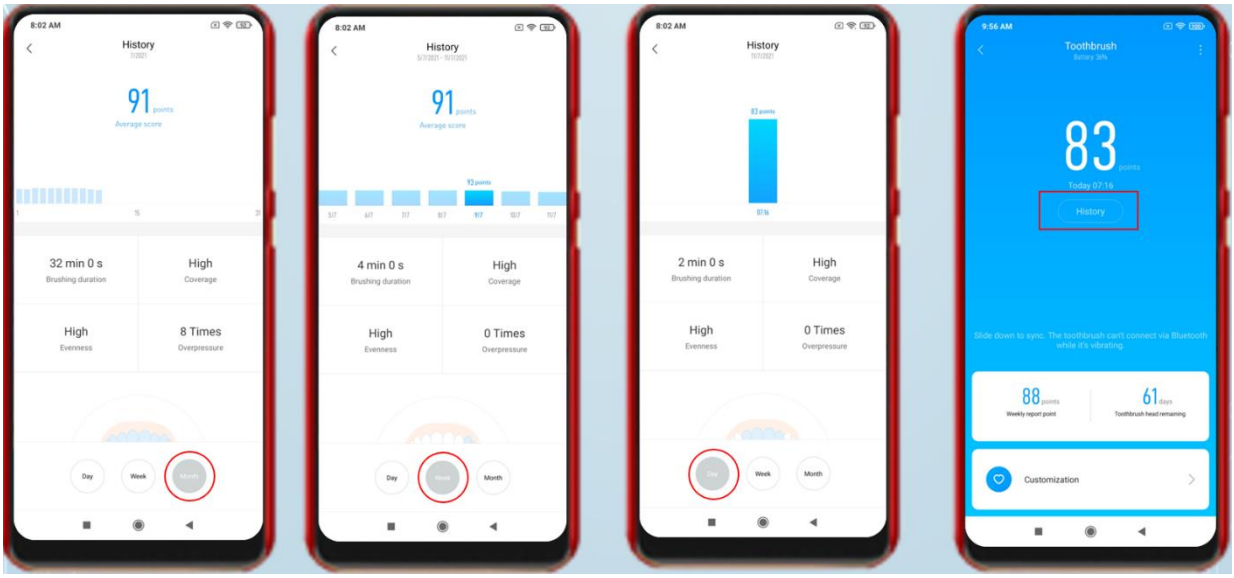

Figure 8 Viewing stored records from the Mi Smart Toothbrush through the app

3.5 Discussion

Without a doubt, the investigators can gain useful insights for any *XIAOMI* smart home under investigation, by analyzing the *Mi Home* application's data. They can demystify the smart home's structure (IoT devices present, etc.), automation scenarios (scenes' configuration, logs, etc.), IoT appliances' logs or/and settings, and even the *XIAOMI* account's information. Therefore, its examination should definitely be part of the IoT digital investigation.

3.5.1 Limitations

Regardless of its valuable takeaways, this research work was subject to several limitations. The evidence collection was restricted to the mobile application. Having collected evidence from other sources (e.g., *XIAOMI* cloud) the findings would be far more fruitful. Finally, the accumulated results of this research work may change partly or completely based on the updates of the application. A recent similar study performed on 2022 [37] proved that some of the results remained the same, while others partially changed.

3.6 Conclusion

To conclude with, this research has yielded some critical artifacts related to the digital investigation of *XIAOMI* IoT ecosystem. More particularly, the *XIAOMI Mi Home* application was forensically explored and various artifacts were uncovered. Utilizing these findings, an investigator can draw certain conclusions about a *XIAOMI* smart home and its IoT appliances. Nonetheless, this work faced some limitations, notably the inability to retrieve other evidence sources like the *XIAOMI* Cloud. These limitations stress the need for further studies and advances so as to fully extract and interpret all the available evidence in such IoT environments. Regardless, considering the goals outlined in **contribution i** (See **Chapter 1** – **section 1.3**), they are regarded as fulfilled.

Chapter 4: Digital Investigation of *DAHUA Technology* CCTV and SECSYS IoT Devices

4.1 Introduction

This chapter includes the forensic examination of *DAHUA Technology CCTV and SECSYS* IoT appliances. *DAHUA Technology* is a Chinese manufacturer of IoT surveillance and security devices. The company also offers a variety of applications, available for multiple OS, which allow remote operation of its products. According to third-party research referenced within its webpage [129], the company has been estimated to be the second-largest supplier of video surveillance equipment in the world since 2014.

4.1.1 Contribution of this chapter

This chapter fulfills the objectives of **contribution ii** (See **Chapter 1** – **section 1.3**). Key takeaways from this research work include:

- The presentation of the way certain *DAHUA Technology* CCTV systems manage their log records, along with methods for examiners to locate and interpret them.
- The presentation of the forensic value of these logs and how they could be leveraged in real cases.
- Assessing the handling of the log records by commercial tools such as *DVR Examiner* [130], *Video Investigation Portable* (*VIP*) [131], *HX-Recovery for DVR&NVR* [132], and *Disk Manager* [133].
- The presentation of the way the *DAHUA Technology* mobile application for the end-user operates.
- The presentation of the artifacts that can be obtained from the forensic analysis of its mobile application and how they could be used in real cases.
- Contributing to FOSS by developing relevant parsers for *ALEAPP* [134] and *iLEAPP* [135] hence enhancing their capabilities utilizing some of the findings of this work.

4.1.2 How this chapter is organized

The rest of this chapter is organized as follows: **Section 4.2** showcases research work related to the log records of *DAHUA Technology* CCTV systems. In **section 4.3**, the mobile application used for remote control of *DAHUA Technology* SECSYS products is examined, and the identified

artifacts are displayed. Discussion about the findings occurs in **section 4.4**, while **section 4.5** concludes this chapter.

4.2 CCTV Systems

Given the pervasive use of CCTV systems, from bustling city streets to serene residences, the digital investigation of these systems is becoming increasingly crucial. This is because they can offer invaluable artifacts and insights that help reconstruct events, pinpoint suspects, and ultimately, aid in resolving crimes more effectively. Nevertheless, the investigation is often intricate due to the myriad of manufacturers, diverse file systems, and the variety of surveillance equipment in use.

File systems are necessary for CCTV devices because they serve as the underlying architecture for organizing, storing, and retrieving their data. These file systems can range from standard file systems to proprietary ones developed by the system's manufacturer. *DAHUA Technology* ships its CCTV products equipped with a combination of file systems. Some of them exclusively utilize its proprietary file system, namely the *Dahua File System* (*DHFS*), others combine DHFS with standard file systems, and still others rely solely on standard file systems.

Li and Zuo [28] studied the DHFS file system in detail and shared their insights on how to recover video footage from it. However, little attention has been paid to the valuable data that are stored within the log records of the file system. Commercial tools have not fully explored this source of evidence yet and the most common methods for accessing it are through the CCTV system's graphical or web user interfaces (GUI/WebUI).

Investigators may typically export these records into text files (using the option labeled as "*Backup*"), and save them either to their own system (if exported through the WebUI) or to a USB stick connected to the CCTV system (if exported through the GUI). Nevertheless, if the access to the CCTV system's GUI/WebUI is protected by an unknown password, and the investigators lack alternative ways to access it, then they might be unable to collect and examine this source of evidence. As a consequence, investigative questions, such as determining the user who modified the CCTV system's configuration to stop recording, might remain unanswered. This research work emphasizes why digital forensic examiners should not overlook this source of evidence and provides means of deciphering meaningful information from it.

In the next section, the equipment and methodology employed are introduced.

4.2.1 Equipment

Five *DAHUA Technology* CCTV surveillance systems were used for this research work: one Network Video Recorder (NVR), one IP camera, and three X Video Recorders (XVRs). XVR is a modern surveillance system that gained popularity in recent years. It enhances the main features of an older Digital Video Recorder (DVR), by adding features of an NVR. The result is a hybrid CCTV system that supports both network and several types of analog cameras.

An analog/IP camera and a hard disk were installed on each CCTV system except for the IP camera where a microSD card was installed instead. Two spare hard disks and a spare microSD were used (as evidence clones) during the experiments. For creating the pictures of the persons enrolled in the "*AI*" menu of the CCTV system "*this-person-does-not-exist.com*" [136] domain was used.

The analysis was conducted on an *Intel i7-9700K Windows 11* (*22H2*) workstation, equipped with 32GB RAM and a *Tableau T3iu* hardware write blocker. For imaging the hard drives *FTK Imager* [137] was utilized and *X-Ways Forensics* [126] was selected for the manual investigation of the evidence collected. As many of the log records were stored in SQLite databases, *DB Browser for SQLite* [127] was selected for viewing their contents and *FQLite* [138] was used for restoring their deleted records. For bypassing password-protected access to the CCTV systems' WebUI, the Google Chrome browser and *DahuaLoginBypass* [139] browser extension were used. The spare devices were cloned using *OSFClone* [140] utility.

The collected images were also inserted into *DVR Examiner* [130], *Video Investigation Portable* [131], *HX-Recovery for DVR&NVR* [132], and *Disk Manager* [133] for automatic analysis. Since some of these tools (e.g., *Disk Manager*) could not parse image files directly, *Arsenal Imager Mounter* [141] was additionally employed to mount forensic images as physical disks whenever deemed necessary. CCTV systems' GUI/WebUI along with a USB stick were also exploited for exporting their log records. The hardware and software used in this work are presented in Tables 3 and 4 respectively.

Table 3 Hardware equipment used in the DAHUA Technology CCTV research work

Hardware	Model/Version
DAHUA Technology XVR with an analog camera	DH-XVR5216AN-4KL-I2
DAHUA Technology XVR with an analog camera	DH-XVR5104HS-I3
DAHUA Technology XVR with an analog camera	DHI-HCVR514C-S3
DAHUA Technology NVR with an IP camera	NVR2108-4KS2
DAHUA Technology IP camera with	IPC-HDBW1435E-W-S2
SanDisk 32GB microSD	SanDisk Ultra

PC workstation with: ▪ Windows 11 ▪ Intel i7 ▪ Tableau Forensic SATA Drive Bay	 22H2 9700K T3iu
SanDisk USB 64GB	Sandisk Ultra

Table 4 Software used in the DAHUA Technology CCTV research work

Software	Version
FTK Imager	4.7.1.2
X-Ways Forensics	20.3 SR-4
DB Browser for SQLite	3.12.2
FQLite	2.0
Google Chrome	111.0.5563.65
DahuaLoginBypass - Chrome extension	4
OSFClone	1.4.1000
DVR Examiner	3.8.0
Video Investigation Portable	21.8.2209.2215
HX – Recovery for DVR&NVR	4.4.9
DAHUA Technology Disk Manager	1.000.0000003.2
DAHUA Technology CCTV system GUI/WebUI	2.680.0000000.22.R – 4.001.0000005.0

4.2.2 Methodology

The methodology followed consisted of three phases namely *Preparation*, *Collection*, and *Analysis*.

4.2.2.1 Preparation

During this phase, the preparation steps of research were taken. The hard drives and the microSD were wiped and inserted into the CCTV systems. The devices were initialized and basic configuration took place. This included setting up system time, choosing detection and operation settings as well as managing systems' users. The CCTV systems' AI capabilities were enabled wherever possible. A dynamic domain service was created to allow remote access to the CCTV systems from outside the local network. This task required a few extra steps regarding network configuration such as enabling port forwarding of specific network ports, etc. Following these actions, the test devices were ready for operation.

The CCTV systems were then used for a period of two months. During this period, interaction with the systems occurred in various manners with the intent of triggering the generation of different log records. Among several actions, the system's GUI/WebUI was accessed both locally and remotely, the system configuration was modified, and both live and recorded footage was viewed.

Furthermore, anti-forensic operations (e.g., format storage) were performed to be able to answer questions that are most likely to puzzle investigators during a real case. The objective

was to determine what happens to the stored logs of the aforementioned CCTV systems when a perpetrator tries to destroy this source of evidence.

Subsequently, the collection phase began.

4.2.2.2 Collection

At this phase, evidence was collected as would happen in a real case. The hard drives and the microSD were removed from the CCTV systems and imaged using *FTK Imager*. The CCTV systems were also equipped with internal memory chips which could not be detached from them, as there was a lack of chip-off equipment. As a consequence, the internal memory of each CCTV system was not examined as part of this research work and remains a subject of future work.

The collected forensic images were cloned using *OSFClone* and the spare devices. This step was taken to be able to export the rest of the log records that were stored within each CCTV system, using its GUI/WebUI, without tampering with original evidence. In particular, clone media were inserted back into the CCTV systems which were powered on. The USB was also attached to each CCTV system when exporting through the WebUI was unavailable. Taking advantage of the GUI/WebUI menu, log records were manually exported into text files (using the option labeled as "*Backup*") and stored within either the connected USB or the workstation PC. Figure 9 shows where the "*Backup*" option appears in the IP Camera's WebUI.

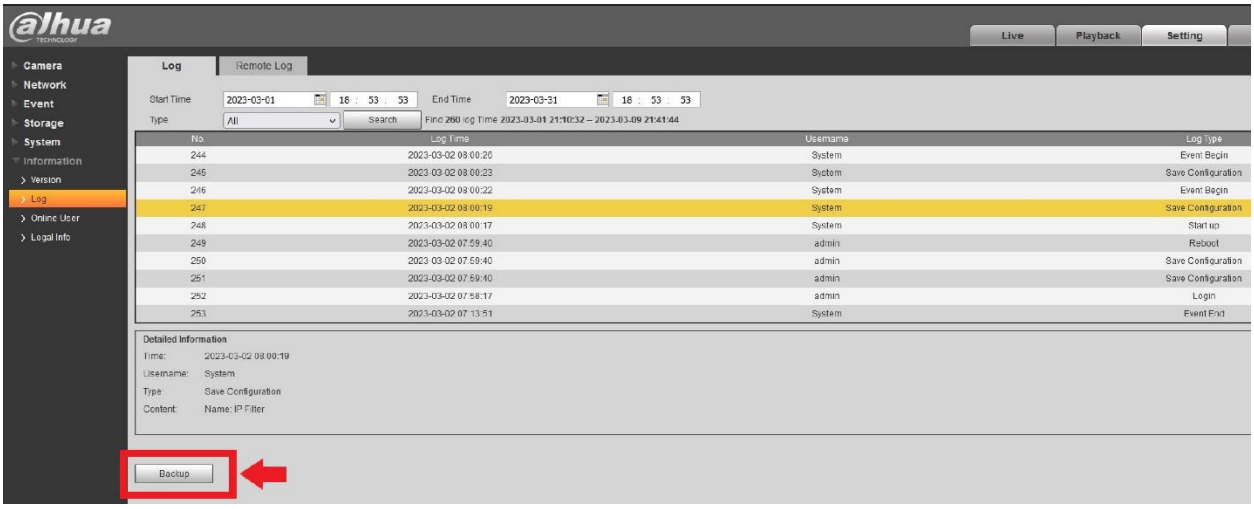

Figure 9 Option to "Backup" log records from one of the DAHUA Technology employed CCTV system's WebUI

Unfortunately, exporting logs are limited to one-month entries per text file. As a result, this action has to be repeated several times to collect all available stored records. The process of

exporting log records was repeated one more time, after removing the hard drives and the microSD from the CCTV systems. This action was taken to check whether any records remained within the system's internal memory instead of their storage media. Having collected previously mentioned evidence sources it was time for the analysis phase.

4.2.2.3 Analysis

At this point, the main analysis of this research work was carried out. The examination begun by investigating where and how the log records get stored. In particular, an effort was made to determine whether each log type resides within each CCTV system's storage media or its internal memory. The location of each log type was successfully identified by inspecting the forensic images, the log records inside the exported text files, and the log entries within the SQLite databases. Subsequently, many of these log records were interpreted. The analysis concluded with the documentation of both findings and insights. *X-Ways Forensics*, *DB Browser for SQLite*, and *FQLite* were utilized for digging into each image and SQLite database.

In addition to the manual analysis, several commercial tools were employed for automatically analyzing the stored log records within the forensic images. However, it appeared that the software utilized did not fully address this type of information.

Having gathered a mixture of results from multiple sources, the findings are shared in the following section.

4.2.3 Results

4.2.3.1 Location of the stored log records

It is very interesting to note that the log records' storage location depends on the type of the CCTV system or the file system setup its storage media utilizes.

To start with, the *DAHUA Technology* IP camera stored all its log records within its internal memory and did not utilize the microSD for their storage. It should be mentioned that the microSD had only one partition, formatted with FAT32 standard file system.

To export its log entries, the IP camera system's WebUI was used along with the workstation. If such systems are password-protected with an unknown passcode, this evidence might be unavailable for further examination unless the company provides a master password that can reset it.

On the other hand, the rest of *DAHUA Technology* CCTV systems stored some of their logs (specific log types) in the hard drives while others (specific log types) were stored within the systems' internal memory. Unless the investigators remove the hard drives from them, all available log records should be included in the text files exported through the CCTV systems' GUI/WebUI, regardless of where they are saved. In Table 5, the location where the logs of each CCTV system employed can be found is listed.

Table 5 The location where each DAHUA Technology CCTV system's log records get stored

CCTV System	Logs stored within the hard disk/microSD	Logs stored within CCTV system's internal memory	Log records available in the exported text files
IPC-HDBW1435E-W-S2 IP camera	None	All	All
NVR2108-4KS2 NVR	Partially	Partially	All
DHI-HCVR514C-S3 XVR	Partially	Partially	All
DH-XVR5216AN-4KL-I2 XVR	Partially	Partially	All
DH-XVR5104HS-I3 XVR	Partially	Partially	All

What is more, the hard disks from these CCTV systems utilized a different file system setup. The disk drives from the NVR and one of the XVRs (DHI-HCVR514C-S3) used one partition, formatted with the proprietary DHFS file system. The rest of the XVR CCTV systems' hard drives used two partitions. The first partition was formatted with the DHFS file system and the second one with the XFS standard file system. Based on the file system setup of each hard drive, the log records were retrieved differently. The file system setup of the storage media of each CCTV system is presented in Table 6. In the following sections the aforementioned file system setups are presented separately, starting with the latter.

Table 6 The file system setup of the storage media of each DAHUA Technology CCTV system

CCTV System	Storage media file system setup
IPC-HDBW1435E-W-S2 IP camera	One partition with FAT32
NVR2108-4KS2 NVR	One partition with DHFS
DHI-HCVR514C-S3 XVR	One partition with DHFS
DH-XVR5216AN-4KL-I2 XVR	Two partitions: 1) One partition with DHFS, 2) One partition with XFS
DH-XVR5104HS-I3 XVR	Two partitions: 1) One partition with DHFS, 2) One partition with XFS

Lastly, interpreting the log entries was a common task for all the employed CCTV systems. Hence, their translation appears in Appendix A.

4.2.3.2 Log records pertaining to CCTV systems whose hard drive utilizes two partitions (DHFS | XFS)

For these XVR CCTV systems, some of their log records were kept in SQLite databases found within the XFS partition while others were located within the internal memory of the CCTV systems. No log records were detected in the partition with the DHFS file system.

A combination of files and folders was found under the root directory of the XFS partition. Its total number varies based on the user's chosen configuration and the CCTV system's capabilities.

If the CCTV system supports "*AI features*" as labeled by *DAHUA Technology*, a folder named "*data*" containing several files and folders would also be located there. "*AI*" was enabled and configured during this research work to explore artifacts related to its usage. These artifacts are displayed in section 4.2.3.2.4.

The files and folders under the root directory are explored in depth in the following sub-sections.

4.2.3.2.1 Main file of interest within XFS partition

The forensically most valuable file within the XFS partition was named "*{Serial Number}_log.db*" (where the *{Serial Number}* was the serial number of the CCTV system). This is the SQLite database where some of the logs were kept. The "*log*" table of this database stored these entries and the "*vlog*" table held extended information for them. The "*timep*" column of the "*vlog*" table records the date of each logged event. The "*modename*" and "*optname*" columns describe the log type and the "*data*" column holds the actual data related to each log entry. Lastly, the "*username*" column saves the name of the user who performed the action recorded in the log. In Figure 10 part of the "*vlog*" table is shown.

	timep	modename	optname	data	username
	Filter	Filter	Filter	Filter	Filter
6	1673986946	Log_Config	Log_SaveConfig	{"Detail":[{"ID":"Enable","Value":...	
7	1673987211	Log_Ugm	Log_Login	{"Log_Time":"2023-01-17 ...	admin
8	1673988602	Log_Config	Log_SaveConfig	{"Detail":[{"ID":"Language","Value":...	
9	1673988757	Log_Config	Log_SaveConfig	{"Detail":[{"ID":"DNS DHCP","Value":["No","Yes"]...	admin
10	1673988757	Log_Config	Log_SaveConfig	{"Detail":[{"ID":"DNS DHCP","Value":["No","Yes"]...	
11	1673988758	Log_Config	Log_SaveConfig	{"Detail":[{"ID":"DHCP","Value":...	admin

Figure 10 Part of the "vlog" table of the DAHUA Technology "{Serial Number}_log.db"

4.2.3.2.2 Where each type of log is stored

After removing the hard drives from these XVRs, their logs were exported into text files. When comparing the contents of these exported text files to the entries of the aforementioned database, the records did not match.

Some types of logs were present only in the exported text files, as they were stored within the internal memory, while others were present only within the database. This indicated that the place where each log is recorded is based on its type. The location where each type resides was determined and is presented in Table 7. A more detailed interpretation of these logs is included in Appendix A.

Table 7 The location where each type of DAHUA Technology log gets stored

Log Type	Log Sub-Type Examples	Logs stored within internal memory	Logs stored within the disk
Playback	Backup Device Found, Search Record	True	False
Storage	Disk, Format, S.M.A.R.T	True	False
System	Reboot, Shutdown, and Sync System Time	True	False
System	Save Config.	False	True
Account	Add User, Delete User, Illegal Login, Modify User, User Logout, User Login.	False	True
Alarm	Illegal Login, Network Disconnection Event, Video Loss, Video Tampering, Intelligent (AI)	False	True
Record Mode	Auto, Manual, Close	False	True
Remote Info	Remote Info	False	True

4.2.3.2.3 Other files of potential interest within the XFS partition

Files of potential interest other than "*{Serial Number}_log.db*" can also be found within the XFS partition. The "*logName.dat*" held the CCTV system's serial number. The "*dataCopyRecord*" listed some of the files that should be found under the root directory of the XFS partition. The "*IoTDataBase.db*", "*PosDataBase.db*", and "*SmdDataBase.db*" were SQLite databases that were found empty during this research work. Nonetheless, these files should also be examined in case they hide any relevant information in other scenarios. Although they were empty, their schemas were evaluated, and the type of information they might store is detailed in Table 8.

Table 8 Files that were found empty in the DAHUA Technology CCTV research work

File name	File type	Potential information it may store
IoTDataBase.db	SQLite database	Information about other IoT devices connected with the CCTV system.
PosDataBase.db	SQLite database	Information related to the usage of POS devices.
SmdDataBase.db	SQLite database	Information related to "Smart motion detection" without utilizing "AI features".

4.2.3.2.4 Files and folders related to "*AI capabilities*" within the XFS partition

One of the XVRs (DH-XVR5216AN-4KL-I2) offered "*AI capabilities*" as labeled by *DAHUA Technology*. These "*AI capabilities*" were related to "*Face Detection/Recognition*" and "*Human/Vehicle Detection*" technologies.

In particular, the CCTV system could detect humans and vehicles as well as recognize faces and notify its user according to their configurations. To be able to accomplish this task, the user needs first to enroll each familiar person (e.g., employee) into the CCTV system's "*AI*" menu by inserting the person's information (gender, age, country, etc.) and uploading a picture of the person's face. Alternatively, the user can utilize the "*human detection*" feature from the "*AI*" menu to choose one of the pictures it detected, for enrolling a person's face. Using this information, the CCTV system's "*AI*" can recognize known individuals (e.g., residents), detecting any unknown individuals (e.g., burglars), and alarming its user respectively.

After enabling and configuring the "*AI*", many SQLite databases were populated and several files got created within the "*data*" folder under the root directory of the XFS partition. In this folder, several interesting artifacts were identified.

Details about the enrolled persons could be recovered from the "*registerFaceIntelligent*" table of the "*registerFaceIntelligent.db*" SQLite database which was placed under the "*\data\appdata\database\FaceRecognition*" directory. In addition, the picture that has been assigned to each person could be located within sub-folders under the same directory and was linked with its entry in the database (See Figure 11).

The "*FaceDatabase*" table of the "*0000_FaceDetectionDataBase.db*" SQLite database recorded the faces that were either detected or recognized by the "*AI*". Part of this table is shown in Figure 12.

Similarly, the "*IVSDatabase*" table of the "*0000_IVSDataBase.db*" SQLite database stored the human/vehicle detection criteria (a.k.a. tripwires) for which the "*AI*" would notify (alarm) the user (See Figure 13).

Lastly, the "*FaceRecognition*" table of the "*0000_FaceRecognitionDataBase.db*" SQLite database combined data related to the faces the "*AI*" successfully recognized from the "*0000_Face DetectionDataBase.db*" and "*registerFaceIntelligent.db*" databases. Part of this table

is displayed below (See Figure 14). These three databases were saved under the "\data\appdata\database\history Database" directory.

All the alarm notifications related to the "AI *features*" are of log type "*Intelligent*" and were also recorded within the "*log*" and "*vlog*" tables of the "*{Serial Number}_log.db*" (See Table 7).

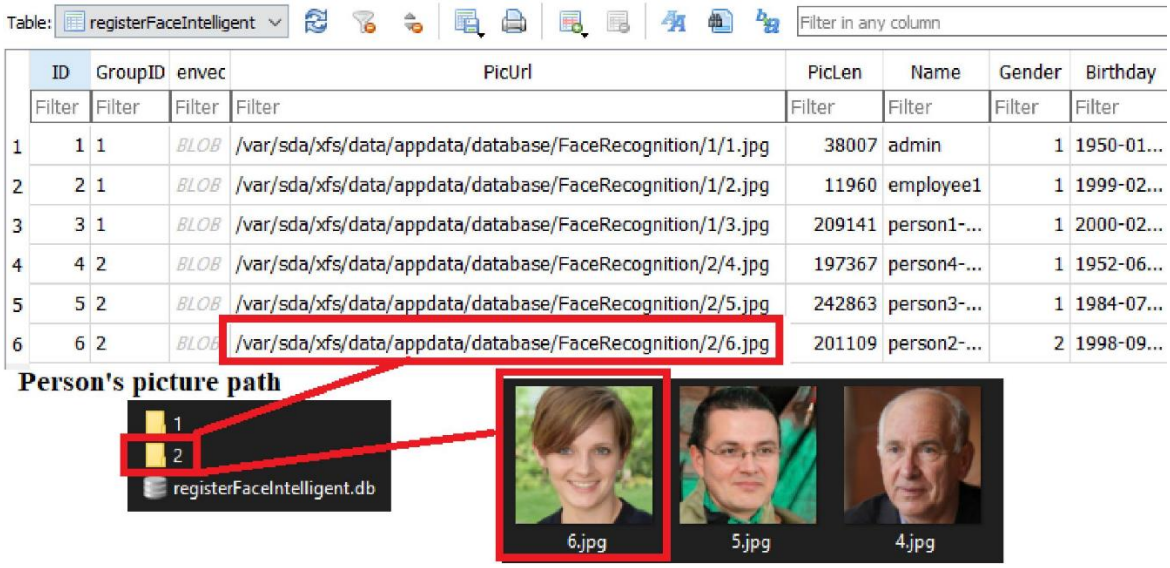

Figure 11 Linking the picture of an enrolled person with its entry in the DAHUA Technology "registerFaceIntelligent.db"

Table: FaceDatabase

	ID	Sex	Age	Emotion	Channel	EventTime	SystemTime
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
7	7	Man	62	10	1	1677105412.0	1677105413.0
8	8	Man	54	10	1	1677105426.0	1677105428.0
9	9	Man	43	10	1	1677105454.0	1677105455.0
10	10	Woman	41	6	1	1677105494.0	1677105495.0
11	11	Woman	33	5	1	1677105546.0	1677105547.0
12	12	Man	46	10	1	1677105623.0	1677105623.0

Figure 12 Part of the "FaceDatabase" table of the DAHUA Technology "0000_FaceDetectionDataBase.db"

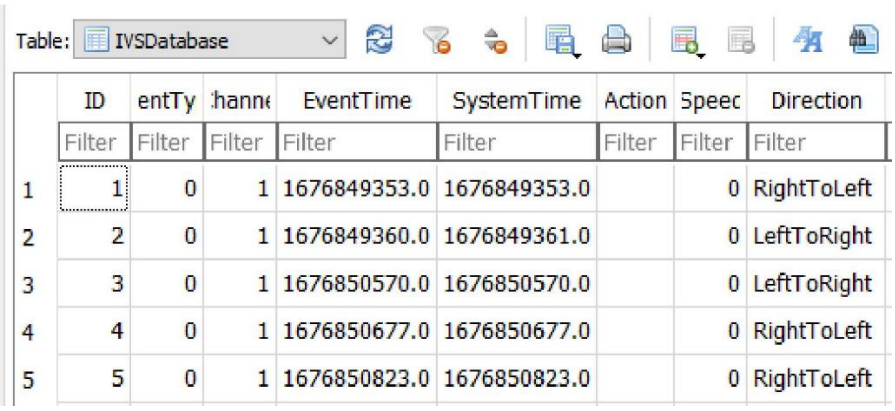

Table: IVSDatabase

	ID	entTy	hanne	EventTime	SystemTime	Action	Speec	Direction
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0	1	1676849353.0	1676849353.0		0	RightToLeft
2	2	0	1	1676849360.0	1676849361.0		0	LeftToRight
3	3	0	1	1676850570.0	1676850570.0		0	LeftToRight
4	4	0	1	1676850677.0	1676850677.0		0	RightToLeft
5	5	0	1	1676850823.0	1676850823.0		0	RightToLeft

Figure 13 Part of the "IVSDatabase" table of the DAHUA Technology "0000_IVSDataBase.db"

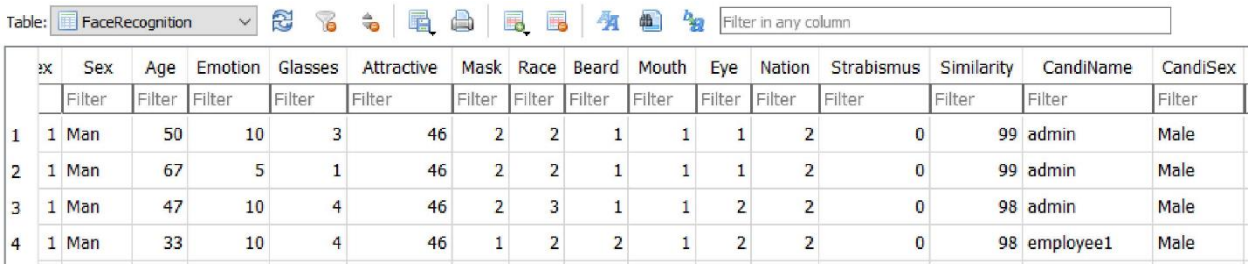

Table: FaceRecognition — Filter in any column

	ex	Sex	Age	Emotion	Glasses	Attractive	Mask	Race	Beard	Mouth	Eye	Nation	Strabismus	Similarity	CandiName	CandiSex
		Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Man	50	10	3	46	2	2	1	1	1	2	0	99	admin	Male
2	1	Man	67	5	1	46	2	2	1	1	1	2	0	99	admin	Male
3	1	Man	47	10	4	46	2	3	1	1	2	2	0	98	admin	Male
4	1	Man	33	10	4	46	1	2	2	1	2	2	0	98	employee1	Male

Figure 14 Part of the "FaceRecognition" table of the DAHUA Technology "0000_FaceRecognitionDataBase.db"

4.2.3.3 Log records pertaining to CCTV systems whose hard drive utilizes one partition (DHFS)

For these CCTV systems, namely the NVR and one of the XVRs (DHI-HCVR514C-S3), a SQLite database containing some of their log records was recovered, making use of the carving capabilities of *X-Ways Forensics*. The rest of their log records were found within their internal memory.

The two carved SQLite databases had exactly the same structure as the aforementioned "*{Serial Number}_log.db*" (See section 4.2.3.2). They also stored the same information with this database. Throughout the remainder of this chapter, each carved database will be referred to as "*carved.db*". Since the data between these databases is identical, they will not be presented again. However, readers can refer to Tables 7 and 9, and Appendix A for details. Unfortunately, since DHFS is a proprietary file system, no further metadata related to these carved SQLite databases (e.g., filename) were found.

Additionally, SQLite databases similar to those mentioned in section 4.2.3.2.4 were also carved. However, as before, they were empty. Based on these findings, it is speculated that depending on the capabilities (e.g., "*AI capabilities*") and the usage of the CCTV systems,

databases like the aforementioned (e.g., those mentioned in section 4.2.3.2.3) might be created and be recoverable through carving.

4.2.3.4 Demystifying anti-forensics artifacts

A user who wants to delete either the video footage or the log records of a *DAHUA Technology* CCTV system like the ones employed in this research work can do it in several ways.

Firstly, the user can format the storage media of the CCTV system since deleting video files and logs individually is not possible. To accomplish this, the user must navigate, using the CCTV system's GUI/WebUI, to the "*Storage/Disk Manager*" pane and select the "*Format*" option (See Figure 15 – Option 1). Even though this option seems to delete recorded video footage, all stored logs remain unaffected by this operation. This included the log records stored within the internal memory of the CCTV systems of this research work as well as the logs found within the SQLite databases.

Secondly, the user can both format the storage media of the CCTV system and clear the log records that reside within the "*{Serial Number}_log.db*" and "*carved.db*" databases, by repeating the previous step after enabling the "*Clear HDD database*" option (See Figure 15 – Option 2). This operation cleared the entries from the *"{Serial Number}_log.db*" and "*carved.db*" databases but did neither delete any of the log records within the CCTV systems' internal memory nor any entries from other important databases (e.g., AI-related). Nevertheless, using *FQLite* the deleted entries from the *"{Serial Number}_log.db*" and "*carved.db*" databases were recovered and evidentiary data were retrieved from them (See Figure 16).

Thirdly, the user can choose to delete all of the CCTV system's log records. This operation requires the user to navigate to the "*Maintain/Log*" pane and select the "*Clear*" option (See Figure 17). This action erased all stored records from both the CCTV systems' internal memory and the "*{Serial Number}_log.db*" and "*carved.db*" databases. Surprisingly, this operation still did not affect any of the databases related to the "*AI features*". What is more, the deleted entries from the "*{Serial Number}_log.db*" and "*carved.db*" databases were still recoverable.

The accumulated results of this section have been summarized in the following Table (See Table 9).

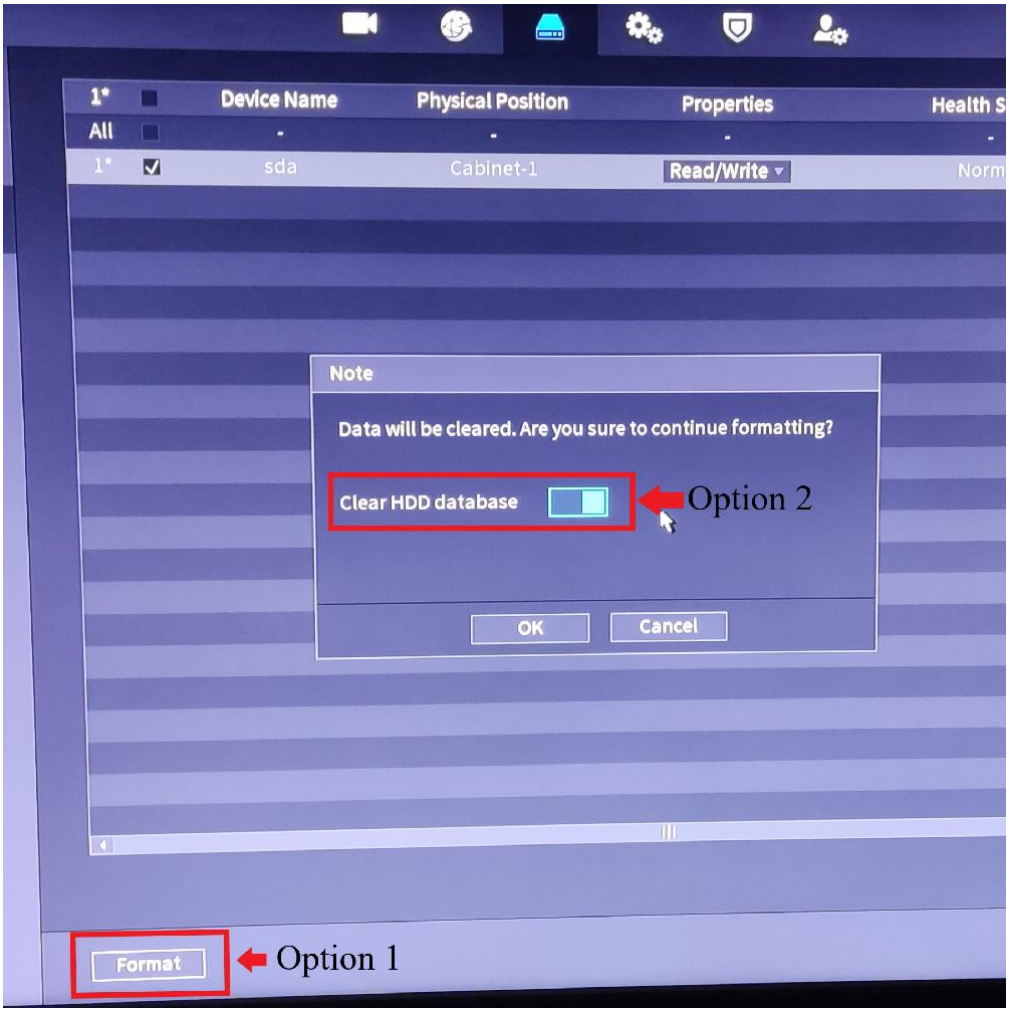

Figure 15 The DAHUA Technology "Format" storage operation' window

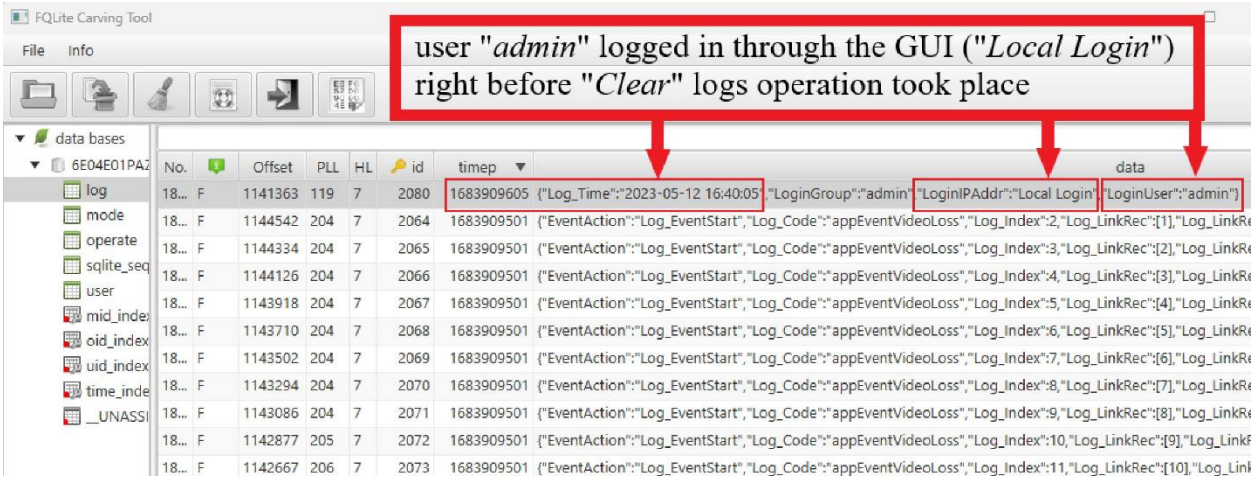

Figure 16 Using FQLite the deleted records from the DAHUA Technology "{Serial Number}_log.db" database were recovered

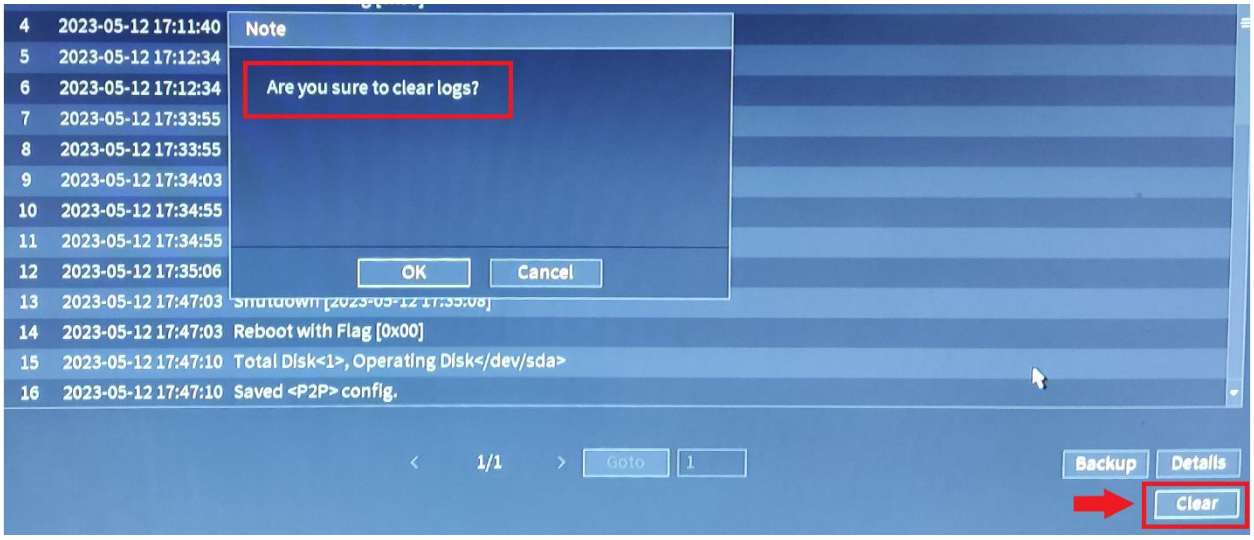

Figure 17 The DAHUA Technology "Clear" logs operation's window

Table 9 How each anti-forensic operation affects DAHUA Technology log records

Anti-Forensic Action Performed	**Logs stored within internal memory**	**Logs stored within the disk ({Serial Number}_log.db and "carved.db")**	**Logs stored within the disk (AI-related databases, etc.)**
Format Storage	Not Deleted	Not Deleted	Not Deleted
Format Storage, Enabled "*Clear HDD database*"	Not Deleted	Deleted/Recoverable	Not Deleted
"*Clear Logs*"	Deleted	Deleted/Recoverable	Not Deleted

4.2.3.5 Bypassing password-protected CCTV systems' WebUI

Dealing with a password-protected CCTV system is rather common in investigations. Getting to know the password of the system under examination is not always the case. Thus, available methods to bypass the password-protection and gain access to their WebUI were explored. *DahuaLoginBypass* Chrome extension was discovered which exploits certain vulnerabilities of the *DAHUA Technology* CCTV systems.

Using this plugin, the authentication prompt for two of the five CCTV systems' WebUI in this research work was successfully bypassed, allowing access. Specifically, access was gained to the WebUI of the IP camera and one of the XVRs (DHI-HCVR514C-S3). Upon accessing their WebUI, all stored logs from these systems were exported. Without this access, this information would have remained unavailable, as these systems store logs within their internal memory.

The reason why this exploit was not successful for the rest of the systems is their system version. System versions that were released after September 2021 have the exploited vulnerabilities patched.

4.2.3.6 Log records' interpretation and forensic value

4.2.3.6.1 Interpretation

The experiments conducted triggered the creation of various log records. The accumulated log records from all aforementioned sources stored the same type of information. In general, this included the user's interaction with each CCTV system, each CCTV system's internal operations, modified configurations, alarms, and more.

Each log record was characterized by a specific log type. Prior to each action, both the date and the type of activity were documented. This facilitated the pairing of each action with its corresponding log entry, both in the exported text files and the SQLite databases. In Appendix A, as many of these match-ups as possible have been documented. Using this source as a reference, examiners may identify actions similar to those described in this research work when they encounter CCTV systems akin to those used here.

4.2.3.6.2 Forensic Value

Without doubt, the analysis of these records could be extremely beneficial for investigators as it would allow them to attribute actions and draw firm conclusions regarding their case. For instance, here are some considerations that may come up during investigations of *DAHUA Technology* CCTV systems:

- The perpetrator or someone else may have formatted the CCTV system's hard drive to delete its stored footage before search and seizure.
- Someone might have disabled the recording of Camera 1 before the specified date.
- Individuals may have accessed either the CCTV system's live footage or playback recordings.
- The perpetrators could have performed these actions either locally or remotely.

The resolution of these considerations and many others depends on the analysis of the stored log records. As a consequence, their evaluation should be an indispensable part of every CCTV system's forensic analysis.

4.2.3.7 Evaluating the commercial tools

After manual analysis, several commercial solutions were tested to assess their efficiency in analyzing this evidence source.

Unfortunately, the utilized tools encountered difficulties in thoroughly recognizing, extracting, and understanding the stored log records. This included *DVR Examiner*, *Video Investigation Portable* (*VIP*), and *HX-Recovery for DVR&NVR* as well as *DAHUA Technology* native application, namely *Disk Manager*. Forensic analysts who make use of these tools in their investigations should be aware of this peculiarity.

4.2.3.8 Insights on how to deal with *DAHUA Technology* CCTV systems

4.2.3.8.1 Locating log records in other *DAHUA Technology* CCTV systems

Based on the findings, the following method is suggested to potentially locate log records in other *DAHUA Technology* CCTV systems.

The investigators should check if the storage media (hard disk/microSD) of the *DAHUA Technology* CCTV system under investigation has one partition, formatted with the FAT32 file systems. If this is the case, then it is highly likely that the CCTV system under interrogation saves its log records within its internal memory.

If the system has one DHFS partition, or has two partitions with one formatted as DHFS and the other as XFS, then its log records are likely stored in both the storage media and its internal memory. Finally, if they discover a different file system or combination of partitions/file systems, they should proceed with caution as this configuration was not met during this research work.

4.2.3.8.2 Prioritizing the collection of this evidence

So far, it has become obvious that the first step investigators need to take when interested in examining these log records should be to ensure access to the system's GUI/WebUI. This should grant them the ability to export and collect all stored logs regardless of this evidence location. However, if access to the system's WebUI is password-protected, using the *DahuaLoginBypass* plugin to access these log records is recommended (See Section 4.2.3.5).

If investigators cannot access the system's GUI/WebUI, these logs might be inaccessible unless the storage media uses one of the file system setups found in this research work. In that case, certain types of logs would still be recoverable (See sections 4.2.3.2 and 4.2.3.3).

4.3 SECSYS

These devices, which also represent a significant portion of the IoT realm, work in tandem to form comprehensive IoT SECSYS. Their primary objective is to protect smart homes, or other locations where they have been installed, and their inhabitants from threats such as intruders and potential hazards like fires or water leaks. Consumers can configure these advanced IoT SECSYS using mobile applications. In the event of an incident, the data stored within these applications can be vital for investigative purposes. Conversely, if inhabitants of the home become the perpetrators, they might attempt to tamper with this source of evidence to cover their tracks and avoid prosecution. In either scenario, the forensic analysis of the application is paramount. Interpreting its stored information can assist in drawing conclusions about the incident under investigation.

In the subsequent section the equipment used and the methodology applied in this work are detailed.

4.3.1 Equipment

Three fresh mobile devices, two *DAHUA Technology* generation 4th CCTV surveillance systems (each one equipped with a camera) and a *DAHUA Technology* kit of IoT security devices were employed for this research. This kit included an alarm hub, a wireless passive infrared (PIR) motion sensor, a wireless door detector and a wireless key fob. This IoT SECSYS was installed into a laboratory. The mobile devices used were a *XIAOMI Redmi Note 6 Pro* with Android 9 (security patch level dated November 2020), a *LG G6* (H870) with Android 9 (security patch level dated May 2019) and an *iPhone X* (A1901) with iOS 15.5. Root access was gained for all devices so as to obtain full file system access. The *XIAOMI* and the *LG* were rooted with *Magisk* [124] whereas the *iPhone* was jailbreaked using palera1n [142].

DAHUA Technology provides a mobile application for the end user of its products which is available for both Android and iOS operating systems. This mobile application is called "*Dahua Mobile Surveillance System* (*DMSS*)" [143]. It is designed to allow the end users to create, configure as well as monitor their IoT SECSYS.

The forensic analysis of this research work was performed on a *Windows 10 Pro* (*21H2*) workstation. *ADB* was used for the majority of data exchange with the Android phones whereas *SSH* was mainly used with the iOS device. Even though utilizing these tools may not be considered as the most forensically sound method to retrieve data from a mobile device, they provided the necessary versatility for the number of conducted experiments. At the end of the

experiments a full file system (FFS) image was acquired from all three devices using *Magnet Acquire* [144] and *libimobiledevice* [145] for the Android and iOS respectively. The purpose of these images was to locate any residual artifacts that could be missed if only *ADB* and *SSH* were preferred.

For the examination of the application's data, *X-Ways Forensics* was utilized [126]. What is more, as the application partially stored data in SQLite databases, *DB Browser for SQLite* [127] was used for viewing this information. *CyberChef* [146] and *Mushy* [147] were also deployed to interpret the contents of certain binary property list (BPLIST) and property list (PLIST) files encoded with *Base64* encoding scheme. The hardware equipment and software used in this work are presented in Tables 10 and 11 respectively. *DAHUA Technology* mobile application along with its versions examined in this research work are listed in Table 12.

Table 10 Hardware equipment used in the DAHUA Technology mobile app research work

Hardware	Model–Version
DAHUA Technology Gen. 4th XVR equipped with an analog camera	DH-XVR5104HS-I3
DAHUA Technology Gen. 4th NVR equipped with an IP camera	NVR2108-4KS2
DAHUA Technology Alarm Hub	DHI-ARC3000H-W2(868)
DAHUA Technology Wireless PIR Detector	DHI-ARD1233-W2(868)
DAHUA Technology Wireless Door Detector	DHI-ARD323-W2(868)
DAHUA Technology Wireless Key Fob	DHI-ARA24-W2(868)
XIAOMI	Redmi Note 6 Pro- Android 9 (SPL November 2020)
LG G6	H870 - Android 9 (SPL May 2019)
iPhone	A1901 (X) – iOS 15.5
PC workstation	Windows 10 Pro (21H2)

Table 11 Software used in the DAHUA Technology mobile app research work

Software	Version
Magisk	23
Palera1n	1.4.0
X-Ways Forensics	20.3 SR-4
ADB (Platform-Tools for Windows)	33.0.3
SSH	OpenSSH_for_Windows_8.1p1, LibreSSL 3.0.2
Magnet Acquire	2.59.0.32716
libimobiledevice	1.3.0
DB Browser for SQLite	3.12.2
CyberChef	9.55.0
Mushy	2.5.0.0

Table 12 Versions of the DAHUA Technology mobile app researched in this research work

Application	Version
DMSS (com.mm.android.DMSS)	Android versions- 1.99.302, 1.99.400, 1.99.401 and 1.99.402
DMSS (com.cesoftware.dmss)	iOS versions - 1.99.302, 1.99.303 and 1.99.400

4.3.2 Methodology

The methodology followed consisted of three phases namely *Reconnaissance*, *Preparation/Collection* and *Analysis*.

4.3.2.1 Reconnaissance

At this stage, familiarization with some of the capabilities of the *"DMSS*" mobile application took place. This step was considered valuable for the purpose of understanding the application's features, forensic artifacts and some of the underlying technologies. The application is available at both Android's and iOS's official repositories (Play Store, App Store). Both Android and iOS applications offered similar capabilities so one of them (Android) is presented.

Initially, end-users can start using the application in two ways. They can either log in to their *DMSS* platform account or simply use the app without an account. *DMSS* platform is offered by *DAHUA Technology* so as to help its customers operate their products. Creating a *DMSS* platform account will allow consumers to bind IoT security devices (e.g., CCTV surveillance systems, Alarm Hubs) to their account and even share their access with other *DMSS* platform accounts. If users choose to create such an account, they will be able to seamlessly access both their bind and shared devices as soon as they log in to their account through the application. On the other hand, if end-users do not want to utilize *DMSS* platform they can start using the app without an account. In either case, the end-users can then take advantage of the app to start adding, configuring and operating their products. Some of the app's capabilities are explained below:

- **Configure Devices**: The users can either add/remove appliances from the IoT SECSYS. They can also configure each IoT device separately based on its functions. For example, for a motion sensor device they can choose its friendly name, the area/room where it is located as well as its detection settings (e.g., entering/exiting delay time, etc.). Another example is a CCTV surveillance system for which they can remotely configure its recording settings (e.g., continuous/scheduled), alarm settings (e.g., which events should trigger alarm notifications) or even remotely format its internal hard drive through the app. If a *DMSS* platform account is used then they would also be able to share access to the aforementioned devices with

other accounts, choosing the level of access for each shared account (e.g., ability to arm/disarm sensors, view notifications, etc.).

- **Configure Account/App**: If a *DMSS* platform account is used then the users can configure their accounts' information (username, etc.) through the app. Regardless of using an account, they can also configure app's settings (e.g., enable/disable app's password protection, etc.).
- **View/Configure Notifications**: The users can view notifications related to their IoT SECSYS. Each IoT device type pushes its own alarm notifications based on its features (e.g., motion detected, etc.) and user's active settings. These notifications are stored in the cloud and are accessible by the users for up to seven days.
- **Live View/Playback**: If a CCTV surveillance system or another similar device (e.g., Video Doorbell) is part of their IoT SECSYS, the users can either access its live footage (Live View) or its stored recordings (Playback). What is more, they can both create/store snapshots and video files from accessed footage. These files are accessible through the app.

Getting a grasp of the mobile application's capabilities was beneficial as this step revealed both the types of forensic artifacts the investigator should look for and the technologies the app takes advantage of. The information exchange between the "*DMSS*" mobile app and the IoT SECSYS formed in this research work is presented in Figure 18.

4.3.2.2 Preparation/Collection

During this phase the preparation steps of research were taken and the evidence to be examined was collected.

To begin with, the CCTV surveillance systems along with the IoT security devices were installed into a laboratory. Next, the application was installed on the mobile devices. For the purpose of the experiments several *DMSS* platform accounts were created. Using the mobile application, a new IoT SECSYS was created. The aforementioned appliances were then assigned to it and configured accordingly. The application was then used for a period of two months in all mobile devices. During that period, multiple actions were performed using the application, including modifying the app's settings, monitoring system notifications, sharing access with other *DMSS* platform accounts, saving footage from the CCTV, and more. After the two-month period the collection phase started.

As a way to identify how the application's features affect forensic artifacts, a dynamic evidence collection process was adopted. Application's data was collected using ADB and SSH commands in parallel with the conducted experiments so as to be able to spot any variations in the artifacts and draw more solid conclusions from its forensic analysis. Application's data were collected more than 60 times in total from all mobile devices. In the following Table (See Table 13) the actions performed prior to each evidence acquisition is demonstrated. At the end of the experiments an FFS image was acquired from mobile devices in pursue of any residual findings outside the application's space. Having gathered the above pieces of evidence the analysis phase started.

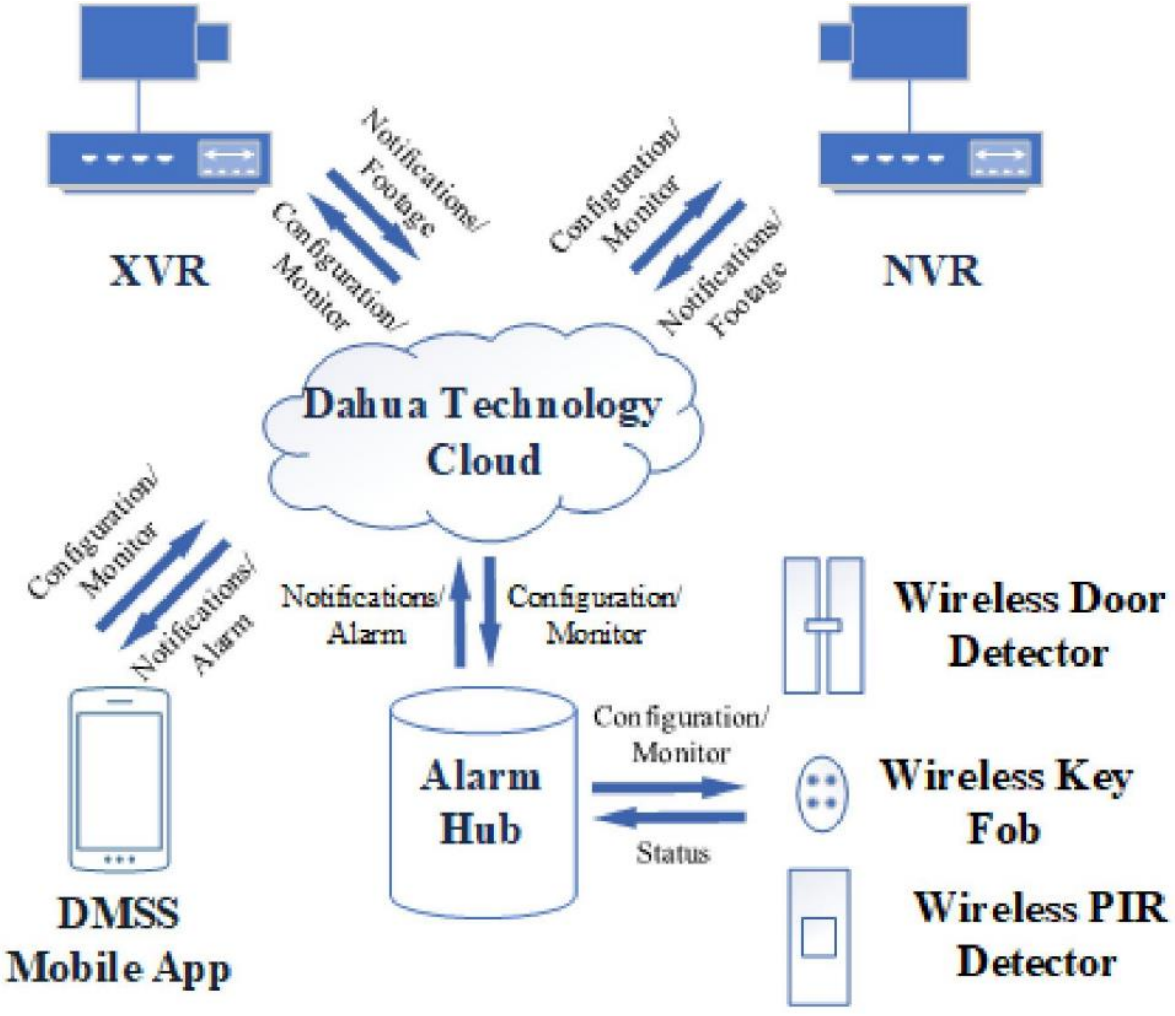

Figure 18 Information exchange between "DMSS" mobile app and the IoT SECSYS of the DAHUA Technology mobile app research work

Table 13 Collected evidence per action performed in the DAHUA Technology mobile app research work

Action Performed	No. of Android App's Collected Evidence	No. of iOS App's Collected Evidence
Install app	2	1
Login/Logout DMSS Platform account	4	2
Add/Edit CCTV/Alarm Hub/Sensors/Key fob	4	2
Share access/ Edit access	4	2
Login/Logout shared DMSS Platform account	4	2
View/Delete notifications	4	2
Edit devices' settings	4	2
View/Edit app settings	4	2
Create/Delete area	4	2
Live View/Playback	4	2
Store snapshots/videos	4	2
Uninstall app	2	1
Total	**44**	**22**

4.3.2.3 Analysis

The main goals of the analysis of the acquired evidence were to identify all potentially valuable artifacts and contribute to FOSS. The findings of the analysis stage are presented in the following section.

4.3.3 Results

Findings are divided in three sub-sections: "*Android application's data artifacts*", "*iOS application's data artifacts*" and "*Contributing to FOSS*".

4.3.3.1 Android application's data artifacts

Table 14 lists all the forensically valuable artifacts that were documented on Android OS. To start with, many files resided under "*/shared_prefs*" directory but only a few of them could be useful to a real case investigation. The first one was "*dss_password.xml*". If a password is used to protect app's access, then this password can be found within this file. Unfortunately, the password was not saved in plain text but in a 32-characters long alphanumerical string which could not be deciphered. Another two files of use that held the region (country code) the user selected while setting the application up were "*is_first_select_country.xml*" and "*is_force_select_ country.xml*". Lastly, another significant artifact found under this directory was "*dh_data.xml*". This file kept track of the latest *DMSS* platform account's email used to log in the app. It also stored a variable named "*USER_NAME_HELP*". The value of this variable (from now on {*userName*}) was an alphanumerical string which was found unique for each *DMSS* platform account. {*userName*} consisted of two parts, a "*random*" string prefix followed by a Unix milliseconds time (e.g.,

ezbt145fdheganrkt1669736325220). This time referred to the date when this particular DMSS platform account was created.

The "*databases*" folder was a goldmine of artifacts for the investigator. Firstly, the examiner should take a look at "*devicechannel.db*" and "*.db*" SQLite databases. The first one gets populated when a CCTV surveillance system is added to the app. Inspecting the "*devices*" table of "*devicechannel.db*" revealed the names ("*devicename*" column) that were given to each CCTV system when added to the app by the user (e.g., NVR-Warehouse). Moreover, "*channels*" table stored the names ("*name*" column) of each of the CCTV system's recording channels (e.g., Front-Door).

If no *DMSS* platform account is used then the "*.db*" database stores the rest of the information related to the formed IoT SECSYS. On the contrary, if a *DMSS* platform account is utilized then this information populates a SQLite database that gets created as soon as the users log in to their account and is named after the {*userName*} value of their account. The "*.db*" and "*{userName}.db*" databases had the same schema and stored the same type of data. These files were considered the most crucial forensically when dealing with this mobile application. By examining the contents of them an investigator could obtain intelligence related to both the devices of the IoT SECSYS (serial number, settings, etc.) and the alarm notifications of the system under question. What is more, if multiple *DMSS* platform accounts log in to the app, then all would have their own "*{userName}.db*" SQLite database to hold their system's data. This is advantageous for investigators as they could also determine historically logged in users and their system's activity. As far as their contents is concerned, "*AlarmPartEntity*" table recorded the configurations (e.g., arming/disarming delay, names, etc.) of each IoT security appliance that was paired with the alarm hub. "*CloudDevices*" table held details for the alarm hub and each of the CCTV systems (e.g., model, name, sharing status, etc.). "*GeneralAlarmMessage*" table cached the latest alarm notifications of the IoT SECSYS. The number of this table's entries varied and depended on how many of these notifications have been retrieved from *DAHUA Technology* cloud prior to the collection of this evidence source. According to *DAHUA Technology* the maximum time period such a notification can be retrieved from its cloud is seven days. Even though all notifications are stored within the "*GeneralAlarmMessage*" table in the backend, when the users access them through the app, they would find them grouped by different channels. This is because each of the IoT SECSYS appliances has its own separate notification channel. Information about these channels was found at the "*CloudChannels*" table.

Lastly, all snapshots and video files created by the user of the app while viewing footage (Live View/Playback) from the CCTV systems were put under "*/sdcard/Android/data/com.mm.android.DMSS/files/Download/snapshot/*" and "*/sdcard/Android/data/com.mm.android.DMSS/files/Download/snapshot/video*" directories respectively. These files were named after the time they were created (e.g., "*20230216074022.jpg*"). The snapshots were stored in "*.jpg*" format whereas the video files were stored in either "*.mp4*" (when recorded by the user) or "*.dav*" (when saved from Playback view) formats.

Table 14 Identified artifacts on the Android OS at the DAHUA Technology mobile app research work

Artifact	Format	Information About
/shared_prefs/ dss_password.xml	XML	-app's passcode.
/shared_prefs/is_ first_select_country.xml	XML	-region selected (country code).
/shared_prefs/is_force_ select_country.xml	XML	-region selected (country code).
/shared_prefs/dh_data.xml	XML	-last logged in DMSS platform account's email and {userName}.
/databases/devicechannel.db	SQLite	-added CCTV system's name and recording channels.
/databases/.db	SQLite	-IoT security system:
/databases/{userName}.db	SQLite	(added/shared devices and settings, notifications, etc.)
/sdcard/Android/data/com.mm.android.DMSS/files/Download/snapshot	folder	-snapshots files stored through the app.
/sdcard/Android/data/com.mm.android.DMSS/files/Download/snapshot/video	folder	-video files stored through the app.

4.3.3.2 iOS application's data artifacts

Table 15 summarizes the discovered artifacts on iOS OS. The "*config.plist*" file located at "*Library**Support*\" directory tracked the previously mentioned {*userName*} value (stored here as the value of the "*tokenUserName*" variable instead) of the latest *DMSS* platform account used to log in the app. It also stored another variable named "*userID*". The value of this variable (from now on {*userID*}) was a numerical string that was found unique for each *DMSS* platform account used. However, when a *DMSS* platform account was not used the value of {*userID*} was equal to "*0*". Under the same directory another interesting artifact was found, "*configFile1*". This file was a bplist file whose contents were encoded with *Base64* encoding scheme. When this file was decoded, the application's password was revealed in plain text (under the key "*5*"). This could aid the examiner in gaining access to the app if an unknown password was set by its user. The last important artifact at this directory was "*Devices.sqlite3*" SQLite database. This file was the equivalent of Android's "*devicechannel.db*". It was slightly differently structured but recorded the same information within "*DEVICES*" and "*CHANNELS*" tables respectively as soon as a CCTV surveillance system was added to the IoT system through the app.

If no *DMSS* platform account is used then the "*DMSSCloud.sqlite*" SQLite database located at "*Library* *Support**0*\" stores the rest of the information related to the formed IoT SECSYS. This file was the equivalent of the aforementioned "*.db*" database. On the contrary, when a *DMSS* platform account is utilized then a new folder gets created under "*Library**Support*\" directory and is named after the account's unique {*userID*} value. Moreover, if multiple *DMSS* platform accounts log in to the app, then all would have their own {*userID*} named folder. Inside the {*userID*} named folder another "*DMSSCloud.sqlite*" database gets created which holds the same information for this particular account. Likewise, this database operated as the aforementioned "*{userName}.db*". The structure between the "*DMSSCloud.sqlite*" and "*.db*"/"*{user Name}.db*" databases was not the same but once again the type of data they stored remained the same. Instead of the above "*AlarmPartEntity*", "*CloudDevices*", "*GeneralAlarmMessage*" and "*CloudChannels*" tables, this SQLite database used "*GatewayPartTable*", "*DEVICES*", "*CHNALARMMESSAGE*" and "*CHANNELS*" tables respectively. Another file that was met while digging through the {*userID*} named folder of each *DMSS* platform account was "*userConfigFile*". This file was a bplist file whose contents were also encoded with *Base64* encoding scheme. After decoding this file, the account's email, nickname and region selected were uncovered.

Last but not least, the snapshots and video files saved by the user of the app while viewing footage from the CCTV systems were put under "*Documents**Captures*" and "*Documents* *Videos*" directories correspondingly. Like before, the naming conventions and file formats of these files were the same.

Table 15 Identified artifacts on the iOS at the DAHUA Technology mobile app research work

Artifact	Format	Information About
\Library\Support\config.plist	BPLIST	-last logged in DMSS platform account's email, {userName} and {userID}.
\Library\Support\configFile1	Base64-encoded BPLIST	-app's passcode.
\Library\Support\Devices.sqlite3	SQLite	-added CCTV system's name and recording channels.
\Library\Support\0\DMSSCloud.sqlite	SQLite	-IoT security system:
\Library\Support\{UserID}\DMSSCloud.sqlite	SQLite	(added/shared devices and settings, notifications, etc.)
\Library\Support\{UserID}\userConfigFile	Base64-encoded BPLIST	-DMSS platform account's email, nickname and region.
\Documents\Captures	folder	-snapshots files stored through the app.
\Documents\Videos	folder	-video files stored through the app.

4.3.3.3 Contributing to FOSS

Code contributions were made to the *ALEAPP* [134] and *iLEAPP* [135] software based on the findings of this research work. Specifically, SQLite queries were developed to recover evidentiary

data from “*devicechannel.db*”, “*.db*”, “*{userName}.db*”, “*Devices.sqlite3*” and “*DMSSCloud.sqlite*” databases. These queries were then integrated into Python parsers to enhance the capabilities of these tools. Both the queries and parsers are now available in the official repositories for *ALEAPP* and *iLEAPP*. An *ALEAPP* report of one of the parser's results is presented below (See Figure 19). This concludes the *“Results*” section of this research.

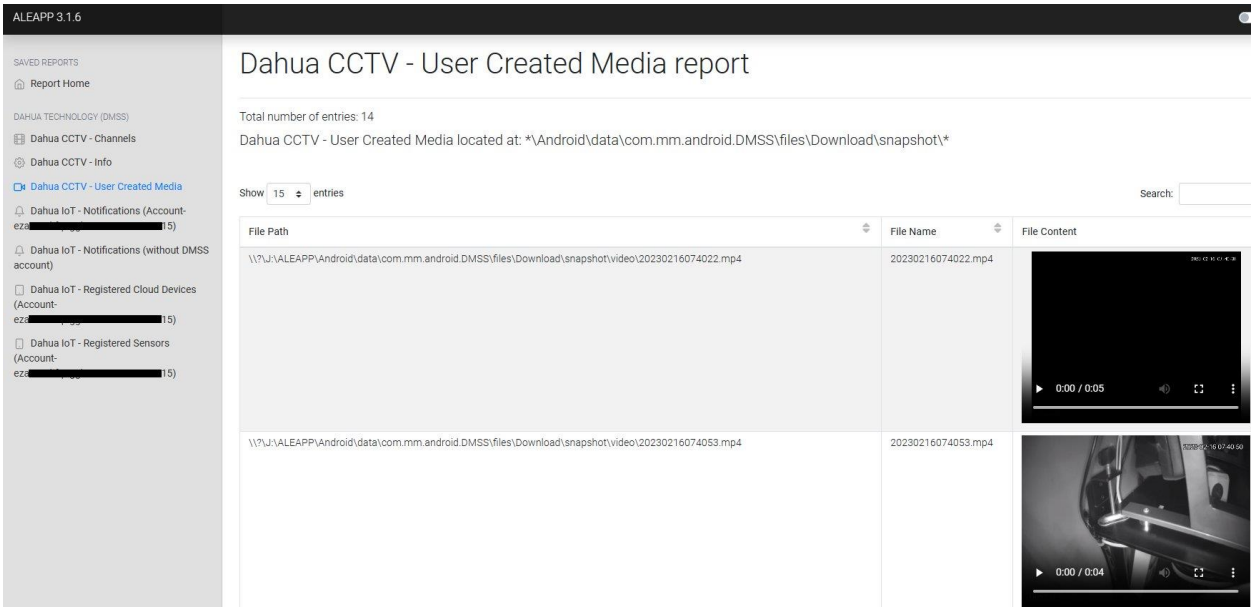

Figure 19 An ALEAPP report of DAHUA Technology parser's results

4.4 Discussion

4.4.1 CCTV Systems

No previous studies regarding the *DAHUA Technology* CCTV systems’ stored log records were found, yet this artifact can be proven pivotal in certain investigations. The methodology employed offers insights into this evidence source. It yielded a plethora of artifacts, leading to a comprehensive forensic evaluation.

Using this research work as a point of reference, forensic examiners who encounter CCTV systems similar to those used in this research may determine where these log records are typically stored, how to manually retrieve them, and how to interpret them. Additionally, the findings demonstrate a technique with which they can access these logs from certain password-protected *DAHUA Technology* CCTV systems’ WebUI. Moreover, methods to detect any anti-forensic actions taken by the suspects are provided.

By exploiting this evidence source investigators may be able to answer a number of important questions such as who and when got access to the CCTV system as well as which actions the logged-in user took. Furthermore, using the results related to "*AI*" usage, they can even probably determine where a particular individual was at a given point in time. These questions could remain unanswered up till now.

4.4.1.1 Limitations

Nonetheless, this research work has its limitations. Despite using five CCTV systems with the intent to potentially trigger a variety of log types, many were never created in this research work. This outcome is attributed to the specific equipment used. CCTV systems and cameras with more advanced capabilities might generate different types of logs or populate the databases detailed in Table 8. Likewise, other types and models of *DAHUA Technology* CCTV systems could potentially store their logs by using/combining different file systems.

Furthermore, due to the absence of chip-off equipment, the internal memory chips of the employed CCTV systems could not be examined. These memory chips store log records and as such it is important to determine their forensic value. Nevertheless, insights into the type of information recorded by these memory chips are shared.

Last but not least, it is worth noting that an update/upgrade in the *DAHUA Technology* systems' version could affect how these logs get stored, how they are recorded, and how they are deleted.

4.4.2 SECSYS

Studies with reference to the forensic analysis of IoT security appliances and their companion applications are rather limited yet the information that gets stored within them could be proven pivotal in certain investigations. Furthermore, no other publicly available research related to the investigation of *DAHUA Technology* applications was found. Using this research work as point of reference along with *ALEAPP* and *iLEAPP* relative results' investigators can identify the structure of the IoT SECSYS under investigation, its registered users/devices, its retrieved notifications and saved footage as well as its current configuration. Doing so they may be able to deal with a number of investigative questions that could remain unanswered up till now. For example, in case of an incident (e.g., burglary) analyzing the latest notifications could indicate when did the incident happened, which appliances witnessed it and more. To the same extent, revealing each registered device's configuration along with the IoT system's structure could help draw more firm conclusions

regarding that incident (e.g., system disarmed during the incident, sensors configured to not operate when system was disarmed, etc.).

4.4.2.1 Limitations

However, this research work is not without its limitations. This work does not take into account the evidentiary data that could reside within the IoT security devices themselves. For example, examining the alarm hub's internal memory could provide supplementary information to the aforementioned. However, this kind of supportive analysis is a subject of future work. Furthermore, the findings of this research work may be partially or fully altered based on updates to the examined mobile application.

4.5 Conclusion

CCTV surveillance systems and SECSYS devices like motion sensors are pervasive IoT products that are designed to protect both public and private spaces. They can also be remotely configured and monitored by their users through applications provided by their manufacturers. Due to their omnipresence, such appliances frequently serve as silent witnesses to criminal activities. Consequently, their digital investigation is crucial for many cases.

DAHUA Technology is a company that produces such IoT devices and offers a wide range of applications that enable their remote operation. However, these applications have not been previously subjected to forensic analysis. Similarly, the log records within its CCTV systems remain uncharted territory. This oversight might leave several investigative questions unresolved. Addressing these questions depends on evaluating the information stored within the log records of *DAHUA Technology* CCTV systems, as well as the mobile application used to control both SECSYS and CCTV devices.

In this chapter, these previously uncharted sources of evidence are explored. By referring to the insights shared here, along with the code contributed to *ALEAPP* and *iLEAPP*, investigators can draw more solid conclusions related to their cases and address questions that previously could remain unanswered. In light of the objectives set out in **contribution ii** (See **Chapter 1** – **section 1.3**), they are deemed achieved.

Chapter 5: Digital Investigation of *HIKVISION* CCTV IoT Devices

5.1 Introduction

Within this chapter, the digital investigation of *HIKVISION* CCTV IoT products is presented. *HIKVISION* is a Chinese company that produces security cameras and other surveillance equipment. Based on a recent analysis from *Research and Markets* [123], *HIKVISION* is considered among the global surveillance camera market leaders. The company offers a wide range of CCTV systems such as DVRs along with a variety of applications, available for multiple operating systems, which can allow remote usage of its products.

5.1.1 Contribution of this chapter

This chapter addresses the goals set out in **contribution iii** (See **Chapter 1** – **section 1.3**). Main insights from this research are as follows:

- The presentation of how the log records are structured within a drive formatted with the *HIKVISION* file system and how an investigator can manually identify them.
- The presentation of the information that gets stored within those logs and how it could assist investigators in real cases.
- Evaluating how commercial solutions like *DVR Examiner* [130], *Video Investigation Portable* (*VIP*) [131], *HX-Recovery for DVR&NVR* [132], and *HIKVISION LocalPlayback* [148] deal with this piece of evidence.
- Contributing to FOSS by developing *Hikvision Log Analyzer* [149], a new open-source utility that can automate the carving and interpretation of these log records.
- The exploration of capabilities that a *HIKVISION* mobile application offer to end users.
- The presentation of the artifacts that can be obtained from the forensic analysis of this mobile application and how they could be used in real cases.
- Contributing to FOSS by developing relevant parsers for *ALEAPP* [134] and *iLEAPP* [135] hence updating their capabilities utilizing some of the results of this work.

5.1.2 How this chapter is organized

In **section 5.2** the research work regarding the digital investigation of the log records of *HIKVISION* CCTV systems can be found. **Section 5.3** includes the research work related to the digital investigation of the *HIKVISION* mobile application that is used for remote control of *HIKVISION* CCTV systems. Discussion about the accumulated results takes place in **section 5.4**, and **section 5.5** wraps up this chapter.

5.2 *HIKVISION* log records

HIKVISION ships its CCTV systems equipped with its proprietary file system, namely the *HIKVISION* file system, which it claims to provide faster indexing and video playback.

Han et al. [29] and Sandeepa et al. [30] comprehensively analyzed and interpreted the HIKVISION file system. They identified its internal structure and presented key areas within the file system. They also demonstrated a process that allowed access to stored video files.

This research work used Han et al.'s analysis as a starting point for navigating through the *HIKVISION* file system. However, its primary focus is on the log records stored within this proprietary file system. It also points out why digital forensic investigators should not overlook this source of evidence and also provides means of deciphering meaningful information from it.

This source of evidence is still unexplored by commercial digital forensic software with the preferred way of accessing it, up to this moment, being through either the CCTV system's graphical or web user interfaces (GUI/WebUI). Things could get worse as questions to an investigation such as when, how, and who got access to the CCTV system may remain unanswered if access to the GUI/WebUI is password protected and the examiner lacks the password or the means of gaining access to it. Even when the investigators get access to the GUI/WebUI, they have to manually export all stored logs to numerous text files as each exported text file can hold no more than 2000 log records. This process can take up a significant amount of time from any ongoing investigation.

In the subsequent section, the equipment and methodology followed are presented.

5.2.1 Equipment

Six *HIKVISION* CCTV surveillance systems were employed in total for this research work: five XVRs and one NVR.

Since the preliminary results of this work indicated there were no major differences between their artifacts, it focused on two of them (one of the XVRs and the NVR). An analog/IP camera and a hard disk were installed on each CCTV system. Another two spare hard disks were used (as evidence clones) during the experiments.

The analysis was conducted on an *Intel i7-9700K Windows 11* (*22H2*) workstation, supplied with 32GB RAM and a *Tableau T3iu* hardware write blocker. To create images of the hard drives, *FTK Imager* [137] was employed, while *HxD* [150] was chosen for the manual examination of the gathered evidence. The hard drives were duplicated using the *OSFClone* [140].

The collected images were inserted into *DVR Examiner* [130], *Video Investigation Portable* (*VIP*) [131], and *HX-Recovery for DVR&NVR* [132] for automatic analysis. CCTV systems' GUI/WebUI and *HIKVISION LocalPlayback* [148] applications were also used for viewing and exporting log records from the hard drives. The hardware and software used in this work are presented in Tables 16 and 17 respectively.

Table 16 Hardware equipment used in the HIKVISION CCTV research work

Hardware	Model/Version
HIKVISION XVR with an analog camera	DS-7104HQHI-K1
HIKVISION XVR with an analog camera	iDS-7204HQHI-M1/S
HIKVISION XVR with an analog camera	DS-7208HQHI-SH
HIKVISION XVR with an analog camera	DS-7216HUHI-K2
HIKVISION XVR with an analog camera	DS-7216HUHI-F2/N
HIKVISION NVR with an IP camera	DS-7616NI-K2/16P
PC workstation with: ▪ Windows 11 ▪ Intel i7 ▪ Tableau Forensic SATA Drive Bay	 22H2 9700K T3iu

Table 17 Software used in the HIKVISION CCTV research work

Software	Version
FTK Imager	4.7.1.2
HxD	2.5.0.0
OSFClone	1.4.1000
DVR Examiner	3.8.0
Video Investigation Portable	21.8.2209.2215
HX – Recovery for DVR&NVR	4.4.9
HIKVISION CCTV system GUI/WebUI	4.0.1
HIKVISION LocalPlayback	3.0.1.2

5.2.2 Methodology

The methodology followed consisted of four phases namely *Preparation*, *Collection*, *Analysis*, and *Verification*.

5.2.2.1 Preparation

During this phase, the initial steps of research were taken. The hard drives were wiped and two of them were inserted into the CCTV systems. The devices were initialized and basic configuration took place. This included setting up system time, choosing detection and operation settings as well as managing systems' users. To allow remote access to the CCTV systems from outside the local network, a dynamic domain service was created. This task required extra steps regarding network configuration such as enabling port forwarding of specific network ports, etc. Following these actions, the test devices were ready for operation.

The CCTV systems were then used for a period of two months. Throughout this duration, various methods of interaction with the systems were employed, aiming to prompt the creation of diverse log entries. Among the different actions taken, the system's GUI/WebUI was accessed both from local and remote points, configurations were changed, and both live and stored recordings were viewed.

5.2.2.2 Collection

During this stage, data was acquired as it would be in an actual scenario. The hard drives were extracted from the CCTV systems and imaged using *FTK Imager*. Later on, these images were cloned using dd and the two spare hard drives. This step was taken to be able to export log records stored within the file system using each system's GUI/WebUI, without tampering with original evidence. In particular, clone drives were inserted back into the CCTV systems which were powered on. Taking advantage of the GUI/WebUI menu, log records were manually exported in text files. Figure 20 shows where the export option appears. Unfortunately, exporting logs are limited to 2000 entries per text file. As a result, this action had to be repeated several times to collect all available stored logs.

5.2.2.3 Analysis

At this point, the main analysis of this research work was carried out. Using findings shared by Han et al. [29] as a starting point and both forensic images and collected text files from the CCTV systems, a heuristic-based examination was performed to determine how the log records get stored within the *HIKVISION* file system. Particularly, an attempt was made to decipher the storage method of each log type by comparing entries in the exported text files with their hexadecimal representation in the forensic disk images. As a result, many of these log records were successfully interpreted. *HxD* was used for manually reviewing each image.

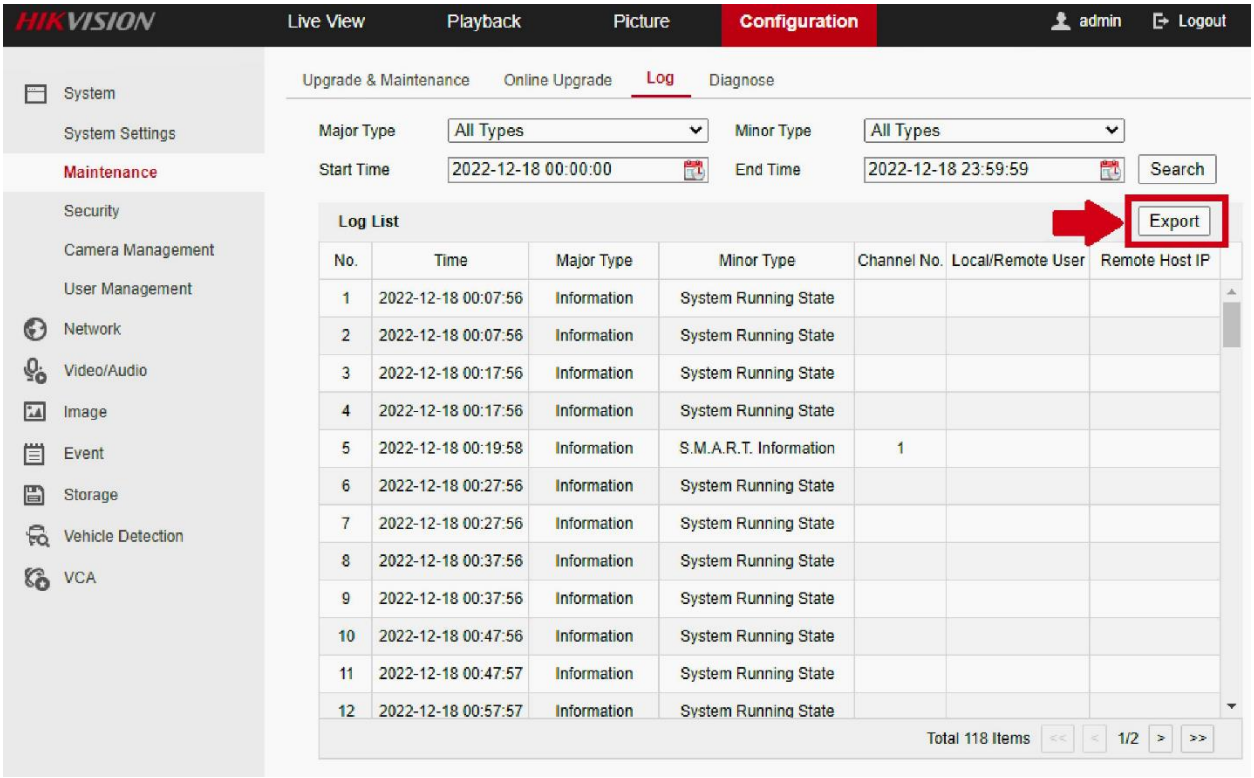

Figure 20 Option to "Export" log records from one of the HIKVISION employed CCTV system's WebUI

In addition to the manual inspection, several commercial tools were employed for automatically analyzing the stored log records within the forensic images. Apart from the *HIKVISION LocalPlayback* utility, it appeared that the commercial tools utilized did not fully capture this type of information. This official *HIKVISION* application provides the ability to view stored logs and recorded footage from a drive formatted with the *HIKVISION* file system. Even though *HIKVISION LocalPlayback* could detect some of the stored log records, its overall accuracy and efficiency could benefit from further refinement.

Given the challenges encountered with the commercial tools used as well as the amount of time that was required to manually analyze each log type, raised the need for a utility that could automate this process. Motivated by this challenge, *Hikvision Log Analyzer* was developed to assist investigators in extracting and interpreting these log records.

After collecting a variety of results from multiple sources, there arose a need to verify the accumulated findings.

5.2.2.4 Verification

The verification phase was the last of this research yet the most important. This was accomplished by comparing the results produced by Hikvision Log Analyzer with the log records from the text files to ensure its proper functionality. Afterward, this process was repeated for the rest of the CCTV systems studied in this research work. The findings of this work are presented in the following section.

5.2.3 Results

5.2.3.1 Log records' structure

The first step was to determine how log records are structured within the *HIKVISION* file system. Even though some initial insights were provided before [29], this research work takes a closer look at how to deal with such data.

The offset to where the log records begin is located at the Master Sector of the disk whose signature is *0x48 49 4B 56 49 53 49 4F 4E 40 48 41 4E 47 5A 48 4F 55* (or "*HIKVISION@HANGZHOU*" in *ASCII*). This decimal number can typically be found at offset "*608* (*0x260* in hexadecimal)", extending for a length of *8* bytes, in *Little-Endian* (*LE*) format. Following at offset "*616* (*0x268* in hexadecimal)", the total size of the log records in bytes can be discovered. This is an *8-byte* length value in *LE*. This piece of information allows the examiner to explore log records' storage area and was exploited when developing *Hikvision Log Analyzer*. Part of a *Master Sector* of one of the drives formatted with the *HIKVISION* file system is shown in Figure 21.

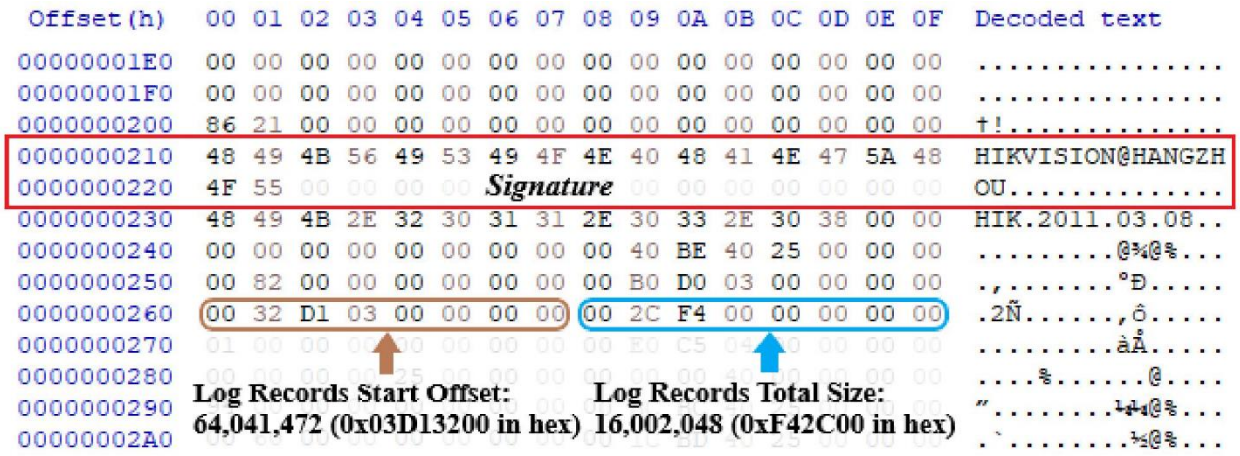

Figure 21 Part of the Master Sector of a drive formatted with the HIKVISION file system

The first *2048* bytes of the log records' storage area were populated with hex values that could not be deciphered. Some of these hex values resembled dates (in "*UNIX seconds*" format) but it was not clear what they represented. Figure 22 shows a segment of these bytes.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
0003D13200	00	00	00	00	00	00	00	00	4C	93	8C	63	01	00	00	00	........L"Œc....
0003D13210	6E	00	00	00	00	00	00	00	E4	A7	00	00	9E	82	6F	63	n.......ä§..ž,oc
0003D13220	A2	A7	6F	63	A3	A7	6F	63	BD	E4	6F	63	44	E9	6F	63	¢§oc£§oc½äocDéoc
0003D13230	96	2F	70	63	96	2F	70	63	B8	96	70	63	68	9B	70	63	–/pc–/pc¸–pch›pc
0003D13240	D7	EE	70	63	D7	EE	70	63	73	36	71	63	CC	3A	71	63	×îpc×îpcs6qcÌ:qc
0003D13250	DE	93	71	63	8E	98	71	63	61	04	72	63	61	04	72	63	Þ"qcŽ˜qca.rca.rc
0003D13260	D8	24	72	63	E6	24	72	63	C1	33	72	63	C1	33	72	63	Ø$rcæ$rcÁ3rcÁ3rc
0003D13270	75	42	72	63	75	42	72	63	C1	56	72	63	C1	56	72	63	uBrcuBrcÁVrcÁVrc
0003D13280	2E	89	72	63	32	89	72	63	1F	EA	72	63	1F	EA	72	63	.‰rc2‰rc.êrc.êrc
0003D13290	9A	5A	73	63	9A	5A	73	63	6C	C6	73	63	1C	CB	73	63	šZscšZsclÆsc.Ësc

Figure 22 Uninterpreted values stored at the first 2048 bytes of the HIKVISION file system's log records' area

At the end of the *2048* bytes, the actual log records began. Their file signature started with *0x52 41 54 53* (or "*RATS"* in *ASCII*) followed by *0x14 00 00 00*. The second part of the file signature was something new as previous studies [29] [30] refer to this value as being *0x01 00 00 00* instead. The value *0x14 00 00 00* was consistent in all images that were examined in this research work. The next *4* bytes were devoted to the system "*Time"* (in "*UNIX seconds"* format - *LE*) when each log record was created. An important point of consideration here is that the "*Time"* value gets stored in the local time zone that the CCTV system was set and not in *Coordinated Universal Time* (*UTC*), regardless of its "*UNIX seconds"* format. An investigator should always remember this peculiarity. Neither previous studies [29] [30] nor the research conducted in this work could determine whether or not the time zone offset was stored within the *HIKVISION* file system.

Following there were *2* bytes that stored the "*Major Type"* of each log and another *2* bytes for the "*Minor Type"* of each log. This information was also new as previous studies [29] [30] simply referred to this part of the log as "*Description"*. The "*Major Type"* and "*Minor Type"* fields are used to specify the type of information each log record stores. From this point on the structure of each log entry differentiates based on the combination of its "*Major Type"* and "*Minor Type"*. The remaining space of each log record was labeled as "*Details"*. Figure 23 presents the structure of log records as it was identified during this research work.

For example, the log records whose "*Major Type"* was "*Operation"* seemed to share the following common "*Details"* structure. The first *16* bytes of the "*Details"* field were used for storing the user's name who performed the logged action. The next *4* bytes were used for saving the user's IP address. Figure 24 illustrates the structure of a log record whose "*Major Type"* was "*Operation"* and "*Minor Type"* was "*Remote*: *Login"*. Again, the remaining information in the "*Details"* field differed depending on the log record type.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
52	**41**	**54**	**53**	**14**	**00**	**00**	**00**	**Creation 'Time'**				**'Major Type'**		**'Minor Type'**	
'Details' of each log															

Figure 23 Structure of log records within the HIKVISION file system

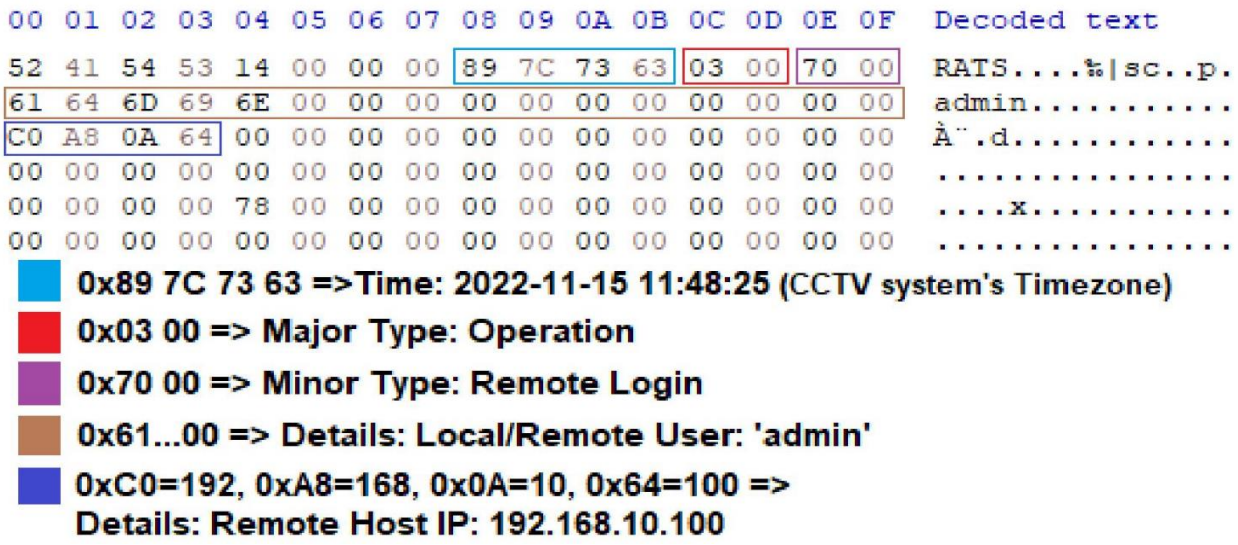

Figure 24 Structure of a log record within HIKVISION file system whose "Major Type" was "Operation" and "Minor Type" was "Remote: Login."

5.2.3.2 Log records' alternative starting offset

The offsets within the *Master Sector* of the *HIKVISION* file system that indicated where the log records began and which was their total size were found identical across all the CCTV systems of this research. In particular, the starting offset was always *"64,041,472* (*0x03D13200* in hexadecimal)" and their total size in bytes was *"16,002,048* (*0xF42C00* in hexadecimal)". However, it was discovered that the log records of the NVR CCTV system began at the offset "*41,472* (*0xA200* in hexadecimal)" regardless of its *Master Sector* suggesting otherwise. Even

though no information within its *Master Sector* that could explain this deviation was found, this offset was also taken into account when developing the utility.

5.2.3.3 Identifying the different types of log records

One of the objectives of this research work was to successfully identify and interpret as many of the stored logs as possible. The first step towards this goal was to examine the combinations of major and minor types of every stored log record. Using the exported text files as a point of reference, *32* different types of log records were successfully mapped with their hex representation. These types are displayed in detail in Appendix B.

Despite having recognized the various major and minor types of the stored logs, unraveling the "*Details*" field of each type was an entirely different task. The main challenge was the amount of information that was stored within the file system for each type of log record. Certain log types used hundreds of bytes for storing their "*Details"* while others only used a few dozen. Even using the exported text files as a map could not assist in overcoming this setback completely. As a consequence, the "*Details"* field for many of the mapped log types was not fully determined. As a rule of thumb, the less information a log type stored, the greater the chances for its complete interpretation. At this point, the need for automating the identification and translation of available log records has emerged.

5.2.3.4 Forensic value of the log records

The *HIKVISION* file system's log records hold information about the user interaction with the CCTV system, the CCTV system's internal operations, and more. Their examination could be extremely beneficial for an investigator as it would allow them to attribute actions and draw firm conclusions regarding their case. Therefore, their evaluation should be an indispensable part of the CCTV system's forensic analysis.

5.2.3.5 Evaluating the commercial tools

Before developing *HIKVISION Log Analyzer*, certain commercial solutions were used to assess their efficiency in analyzing this particular source of evidence.

Several of the tools deployed, including *DVR Examiner*, *Video Investigation Portable* (*VIP*), and *HX-Recovery for DVR&NVR* faced challenges in fully identifying, retrieving, and interpreting the *HIKVISION* file system's log records.

Conversely, the *HIKVISION LocalPlayback* utility demonstrated some ability to parse these log records. However, even this native *HIKVISION* application had room for improvement, as it was able to recover only a very small fraction of the total stored log entries (less than *1%*). For example, as can be seen in Figure 25, the *HIKVISION LocalPlayback* utility found only *114* log records after scanning one of the collected images of this research. On the contrary, when searching for the total occurrences of log record signatures (using the case-sensitive search keyword "*RATS"*) on the same image using *HxD*, the tool reported *19,063* stored logs (See Figure 26). After manually going through them, none of the *19,063* reported records that matched the search keyword "*RATS"* were found to be false positives. Regardless, there might be false positives in other scenarios using just this search keyword. In any case, this difference suggests that *HIKVISION LocalPlayback* missed the majority of the stored logs and therefore should not be trusted for their examination.

No other application was found that could assist with the automatic recovery and translation of the stored log records without the need for manual export through the GUI/WebUI of the CCTV systems. Motivated by this challenge, the *Hikvision Log Analyzer* was developed.

5.2.3.6 Contributing to FOSS

Integrating the accumulated findings from the previous sections, *Hikvision Log Analyzer* is a *Python* utility that was developed to carve available log records from a raw image (extensions *.dd* and *.001* are currently supported) as well as parse exported text files from a *HIKVISION* CCTV system as long as its hard drive is formatted with the *HIKVISION* file system.

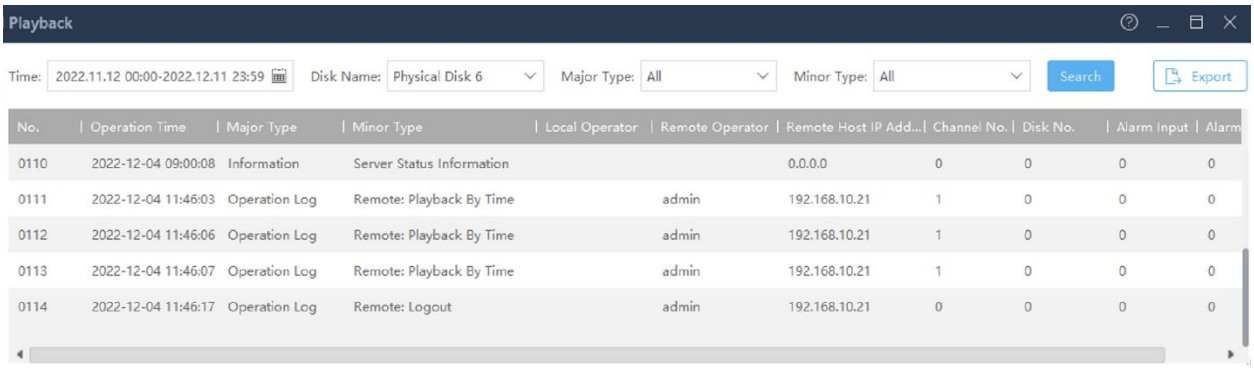

Figure 25 HIKVISION LocalPlayback utility reported 114 stored log records after scanning one of the collected images of the HIKVISION CCTV research work

If a log type is unsupported the utility will still retrieve its "*Major Type*", "*Minor Type*", and "*Time*" fields but will not retrieve any "*Details*" information. In cases where the log type is "*partially parsed*", this application will recover the aforementioned fields along with partial "*Details*" information. The parsing status for each of the identified log types is demonstrated in Appendix B. Figure 27 presents *Hikvision Log Analyzer's* main application window.

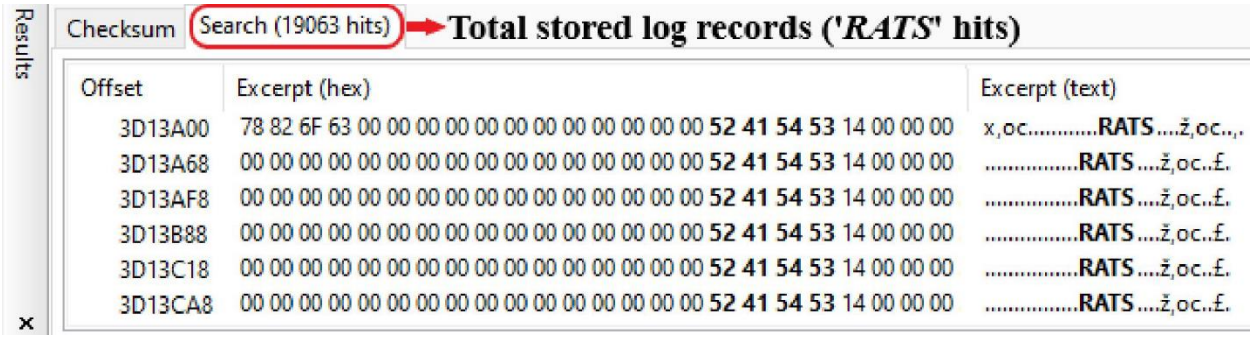

Figure 26 HxD discovered 19.063 stored log records after scanning the same image of the HIKVISION research work

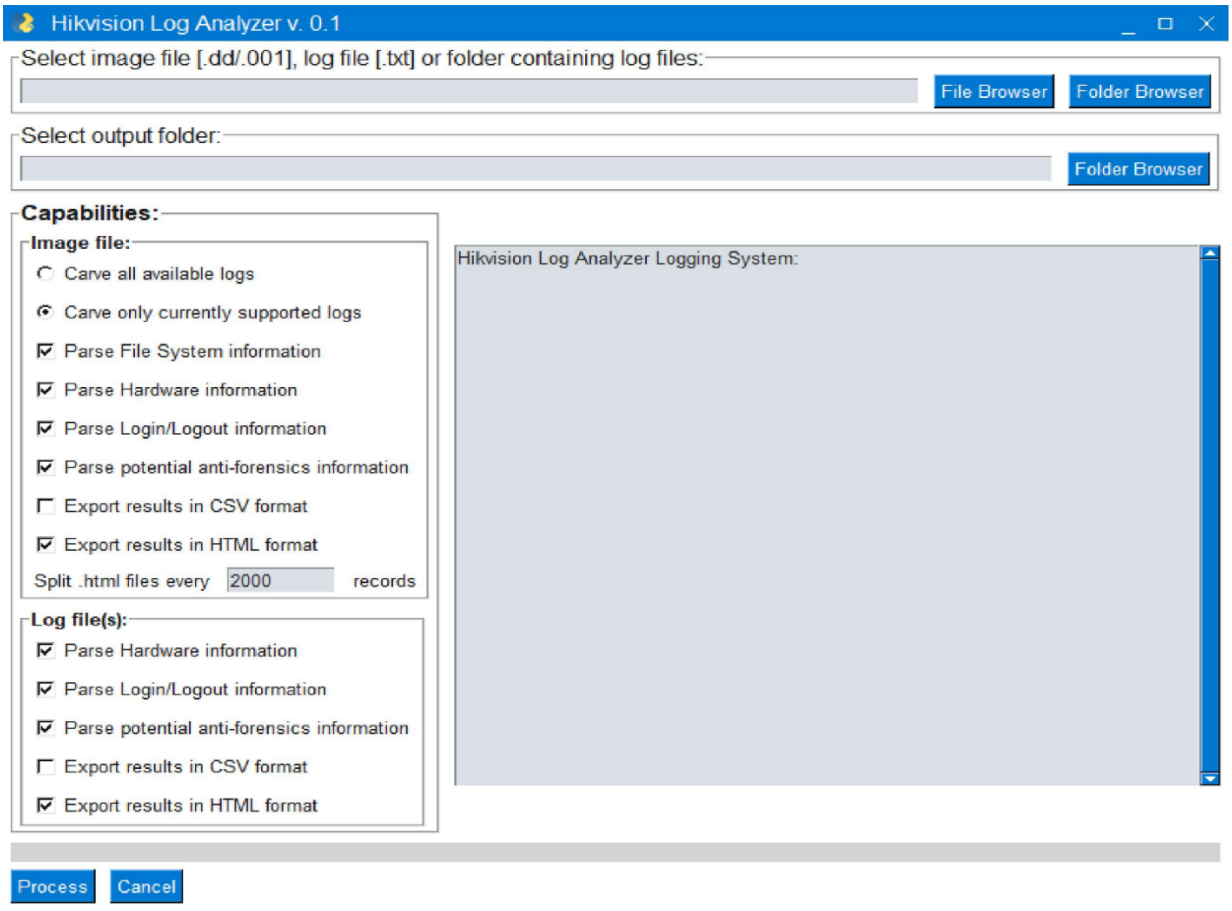

Figure 27 Main application window of the Hikvision Log Analyzer

As illustrated in Figure 27, this utility is also capable of grouping identified log records based on preconfigured categories (e.g., "*Logon Information"* groups *Login* and *Logout* log types). Results can be exported in both *CSV* and *HTML* formats. For instance, in Figure 28 an *HTML* report of "*Logon Information"* is presented. It should be noted that if the CCTV system's GUI/WebUI was password protected with an unknown password it would be difficult to recover this critical information if not impossible. *Hikvision Log Analyzer* can retrieve this information with ease and independently of password knowledge.

When *Hikvision Log Analyzer* scanned the same image that was analyzed with the employed tools (See Figures 25 and 26), it managed to both identify all available log records (*19,063* in total) and to successfully interpret either fully or partially about 75% of them (See Figure 29).

Last but not least, to make it easier for investigators to verify the application's reported results, the current parsing status and the specific offset of each carved log record are also included within its report (See Figure 29). In addition, taking advantage of this feature, an examiner can navigate directly to a log record of interest, and in case its type is unsupported, they could even help map or decode this specific log type. More details about this tool can be found on its *GitHub* page [149].

Number of Log Record	Time	Major Type	Minor Type	Channel No.	Local/Remote User	Remote Host IP	Details	Parsing Status	Entry Offset
180	2022-11-15 21:50:33	Operation	Remote: Login		admin	192.168.10.100		Parsed	65097628
181	2022-11-15 22:24:08	Operation	Remote: Logout		admin	192.168.10.100		Parsed	65101944

Figure 28 Part of Hikvision Log Analyzer "Logon Information" HTML report

There are currently 19063 records within the image file provided.
The tool identified successfully the total number of stored records

Fields that can be used for validation of reported results

Home Page 1 2 3 4 5 6 7 8 9 10

Number of Log Record	Time	Major Type	Minor Type	Channel No.	Local/Remote User	Remote Host IP	Details	Parsing Status	Entry Offset
1	2022-11-12 11:25:18	Operation	Remote: Initialize HDD		admin	192.168.10.100		Partially Parsed	64043520

Figure 29 Part of Hikvision Log Analyzer "Generic" HTML report

5.3 *HIKVISION* mobile app

HIKVISION offers various mobile applications for both Android and iOS platforms [151]. Two of the applications developed to work with CCTV systems which are still maintained and updated, are "*Hik-Connect - for End user*" and "*HiLookVision*".

These mobile applications are designed to allow the end user to operate a CCTV system remotely. The application of choice for this research work was the first one. The criteria were that "*Hik-Connect - for End user*" was more regularly and recently updated. Nonetheless, a brief examination of the "*HiLookVision*" structure was undertaken. It appears that the way both applications store their data is very similar. However, a complete forensic analysis of "*HiLookVision*" application remains a topic of potential future work.

This source of evidence is still unexplored by both commercial and open-source digital forensic software. Things could get worse as critical questions such as when, how, and who got access to the CCTV system as well as which actions the user took using the mobile applications may remain unanswered.

In the subsequent section the equipment used and the methodology applied in this work are detailed.

5.3.1 Equipment

For this research work, three fresh mobile devices were used along with two new *HIKVISION* Generation 4th XVRs equipped with analog/IP cameras. The mobile devices utilized were a *XIAOMI Redmi Note 6 Pro* with Android 9, a *Samsung SM-J10FN* with Android 7.1.1, an *LG G6* with Android 9, and an *iPhone X* with iOS 15.5. The XVRs used were a *DS-7104HQHI-K1* and a *DS-7216HUHI-K2*.

Given that the initial findings of this work suggested minimal variances between their artifacts, it focused on two of the mobile devices (*LG* and *iPhone*) and one of the XVRs.

Root access was gained for both mobile devices in order to obtain full file system access. The *LG* device was rooted with *Magisk* [124] whereas the *iPhone* was jailbreaked using *palera1n* [142].

The analysis was conducted on a *Windows 10 Pro* (*21H2*) workstation. *ADB* was used for the majority of data exchange with the *LG* phone whereas *SSH* was mainly used with the *iPhone*. Even though these tools may not be considered as a forensically sound method to retrieve data from a piece of evidence and either commercial tools or more appropriate methods would most probably be used in a real investigation, this method offered the necessary versatility for the number of conducted experiments. Ways of extracting an FFS or a physical image from a mobile device are out of the scope of this work.

Having said that, at the end of the experiments, an FFS image was acquired from both devices using *Magnet Acquire* [144] for the Android and *libimobiledevice* [145] for the iOS device. The purpose of these images was to locate any residual artifacts that could be missed if only *ADB* and *SSH* were preferred.

For the examination of the application's data, *X-Ways Forensics* [126] was utilized. Additionally, as the application partially stored data in SQLite and realm database formats, *DB Browser for SQLite* [127] and *Realm Studio* [152] was used for viewing this information.

Frida [153], *fridump3* [154], and *CyberChef* [146] were also deployed to retrieve the application's RAM and search for realm databases' decryption keys.

The hardware and software used in this work are presented in Tables 18 and 19 respectively. *HIKVISION* mobile application along with its versions examined in this research work are listed in Table 20.

Table 18 Hardware equipment used in the HIKVISION mobile app research work

Hardware	Model/Version
HIKVISION Gen. 4th XVR	DS-7104HQHI-K1
LG G6	H870 - Android 9 (SPL May 2019)
iPhone X	A1901 – iOS 15.5
PC workstation	Windows 10 Pro (21H2)

Table 19 Software used in the HIKVISION mobile app research work

Software	Version
Magisk	23
Palera1n	1.4.0
X-Ways Forensics	20.3 SR-4
ADB (Platform-Tools for Windows)	33.0.3
SSH	OpenSSH_for_Windows_8.1p1, LibreSSL 3.0.2
Magnet Acquire	2.59.0.32716
libimobiledevice	1.3.0
DB Browser for SQLite	3.12.2
Realm Studio	13.0.2
Frida	16.0.7
fridump3	-
CyberChef	9.55.0

Table 20 Versions of the HIKVISION mobile app researched in this research work

Application	Version
Hik-Connect - for End user (com.connect.enduser)	Android versions- 5.0.0.1125, 5.0.1.1207 and 5.0.2.1213
Hik-Connect - for End user (com.hikvision.hikconnect)	iOS versions - 5.0.0, 5.0.1 and 5.0.2

5.3.2 Methodology

The methodology followed consisted of three phases namely *Reconnaissance*, *Preparation/ Collection*, and *Analysis*.

5.3.2.1 Reconnaissance

At this phase, familiarization with some of the features of the "*Hik-Connect - for End user*" mobile application's capabilities took place. This task was essential in pursuance of understanding the application's complex features, its artifacts, and some of the *HIKVISION* technologies.

The application is available at both Android's and iOS's official repositories (Play Store, App Store) as well as at *HIKVISION* app repository (Hikvision App Store). According to Play Store's statistics [155], the Android application has surpassed *5* million downloads worldwide. Both Android and iOS applications offered similar capabilities so one of them (Android) is explained.

To begin with, users can start using the application in different ways. They can either log in to their *Hik-Connect* account, start using the app in "*Visitor Mode*" or simply use the app without any type of account.

Hik-Connect is a *HIKVISION* platform designed specifically to help customers operate its security products [156]. Creating a *Hik-Connect* account will allow users to bind security/IoT devices to their accounts and even share their access with other *Hik-Connect* accounts. Of course, binding a security device with an account requires extra configuration steps from within the CCTV system's settings. For example, for such an operation it is mandatory to enable the CCTV system's access to *Hik-Connect* platform.

If users choose to create an account, they will be able to seamlessly access both their bind and shared devices as soon as they log in to such a *HIKVISION* application. On the other hand, if users opt for "*Visitor Mode*" the application will create a local dummy user for them whereas if they do not select either "*Account Login*" or "*Visitor Mode*" they would start using the app without any account.

Apart from utilizing a *Hik-Connect* account, users who want to gain access to a CCTV system through the application can do so in many ways.

They can add it to the app by retrieving its basic information automatically provided that the mobile device and the CCTV system are connected to the same LAN network (option "*Online*

Device"), by scanning its QR code (option "*Scan QR Code*"), or by configuring it themselves (option "*Manually Adding*").

The application currently supports the following types of manual addition:

- **Hik-Connect Domain**: This type requires the CCTV system to already have access to *Hik-Connect* platform. The users need to insert the device's serial number and verification code (both accessible from the CCTV systems' menu).
- **IP/Domain**: This type allows the user to input either the system's local/remote IP or its custom domain. For all of these sub-options, the user must also supply the credentials of the desired CCTV system's user.
- **Pyronix**: *Pyronix* is another manufacturer of security systems and technologies [157]. This type could not be examined further as access to *Pyronix* services was required.
- **Router**: This type is used when the CCTV system belongs to specific models of *HIKVISION* NVRs. This type could not be examined further due to lack of such a device.

Using these methods, users should be able to both remotely view CCTV cameras' live footage and access stored recordings. They also have the ability to create screenshots and videos from the footage they are viewing.

Nevertheless, if they want not only to access the CCTV system but also to be able to configure it, they need to enable remote configuration through the application. Once enabled, the remote configuration will allow users to perform certain actions such as modifying the cameras' recording schedule.

All the identified ways with which users can access and configure a CCTV system through the *HIKVISION* mobile application are summarized in Figure 30.

5.3.2.2 Preparation/Collection

At this phase, the preparation steps of research were taken and the evidence to be analyzed was collected.

To start with, the CCTV system was initialized and basic configuration took place. This included setting up system time as well as creating the system's users.

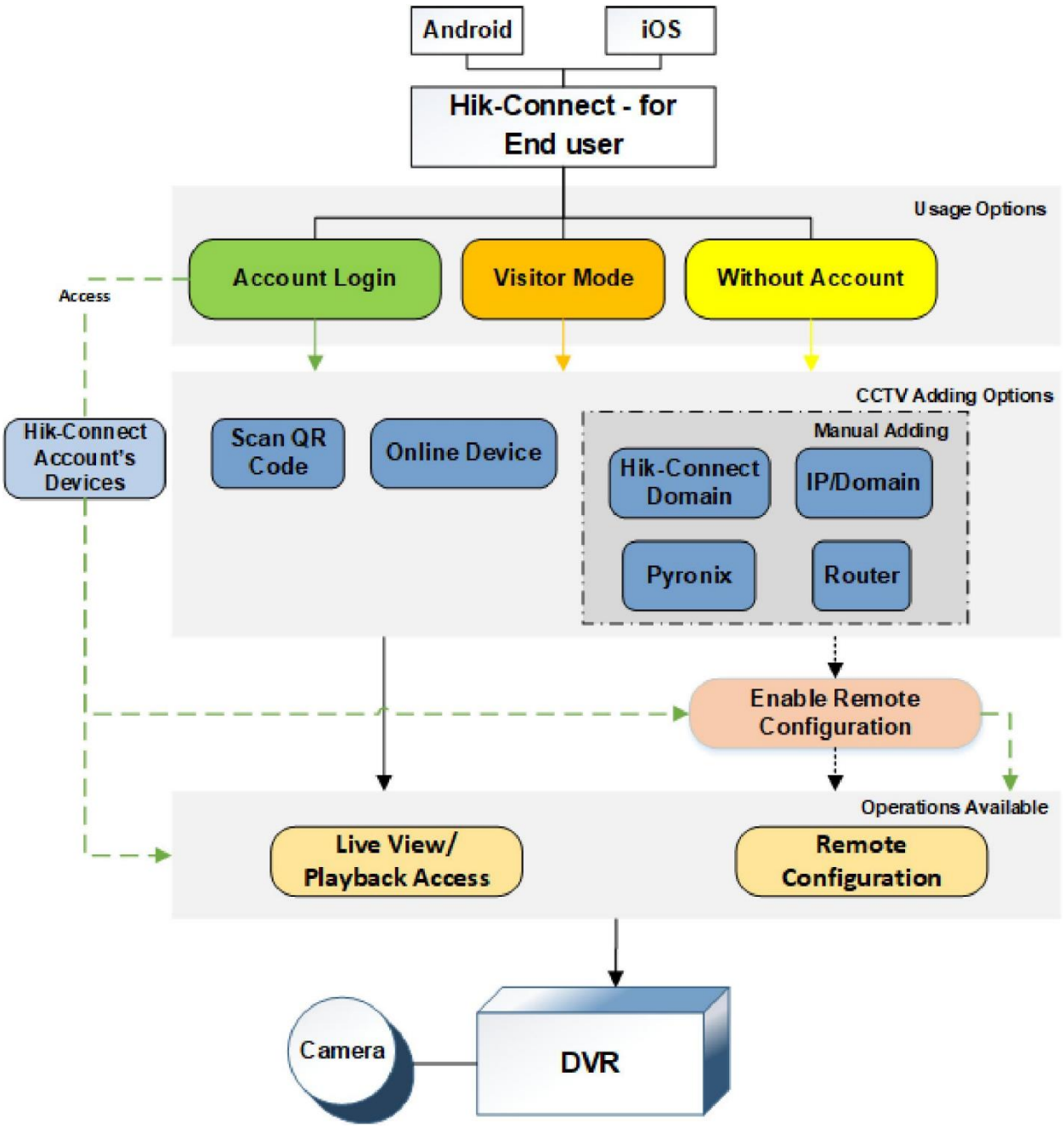

Figure 30 Ways of accessing and configuring a CCTV system through the HIKVISION mobile app

In order to facilitate remote access to the CCTV system from outside the LAN network and test the "*IP/Domain*" type of manual addition, a custom domain service was utilized (*DynDNS* [158]). This task required binding the CCTV system with the domain created. It also required network configuration such as enabling port forwarding of specific network ports at the router. In addition, for the sake of testing the "*Hik-Connect Domain*" type a couple of "*Hik-Connect*"

accounts were created. For the same purpose, the CCTV system's access to *Hik-Connect* platform was enabled among other steps. If this particular setting was disabled, adding a CCTV system with either "*Scan QR Code*" or "*Hik-Connect Domain*" options would be impossible.

The application was then installed on the mobile devices. In both mobile devices, the application was used for a period of nearly two months. During that period diverse actions were performed using the application such as accessing the CCTV system's live footage and stored recordings, configuring it, saving its footage, and more. After the two-month period, the analysis phase started.

The diverse app features forced the adoption of a dynamic evidence collection process. The application's data was collected using *ADB* and *SSH* commands in parallel with the experiments so as to be able to identify any differences in the artifacts and draw more solid conclusions from its forensic analysis. Furthermore, the application's RAM was also collected using *Frida* and *fridump3* during the experiments in favor of decrypting the application's realm databases whenever deemed necessary.

Application's data and RAM were collected more than *80* times in total from both Android and iOS mobile devices. The actions performed prior to each evidence collection are demonstrated at Table 21. At the end of the experiments, an *FFS* image was acquired from both mobile devices in pursuit of any residual artifacts outside the application's space. Having gathered all the necessary pieces of evidence the analysis phase began.

Table 21 Collected evidence per action performed in the HIKVISION mobile app research work

Action Performed	**No. of Android App's Data/RAM Evidence**	**No. of iOS App's Data/RAM Evidence**
Install App	1 Data	1 Data
Login/Logout to Hik-Connect Account	2 Data + 2 RAM	2 Data + 2 RAM
Add CCTV-Scan QR Code	2 Data + 2 RAM	2 Data + 1 RAM
Add CCTV-Online Device	2 Data + 2 RAM	2 Data + 1 RAM
Add CCTV-Manual Adding-Hik-Connect Domain	3 Data + 3 RAM	3 Data + 2 RAM
Add CCTV-Manual Adding-IP/Domain	4 Data + 4 RAM	4 Data + 3 RAM
Access CCTV-Live View	3 Data + 1 RAM	3 Data + 1 RAM
Access CCTV-Playback	3 Data	3 Data
Access CCTV-Create Screenshot	2 Data	2 Data
Access CCTV-Save Video	2 Data	2 Data
Config. CCTV-Disable/Enable Recording	3 Data + 1 RAM	3 Data
Config. CCTV-Time Sync.	2 Data	2 Data
Uninstall App	1 Data	1 Data

Total	30 Data + 15 RAM	30 Data + 10 RAM

5.3.2.3 Analysis

The main objectives of the analysis of collected evidence were to identify all potentially valuable artifacts, verify actions performed by the user of the app, determine how the application handles these artifacts, and contribute to FOSS. The outcomes of the analysis phase are presented in the next section.

5.3.3 Results

5.3.3.1 Artifacts

Artifacts are divided into four sub-sections based on the underlying OS and evidence source: "*Android app's data artifacts*", "*Android app's RAM artifacts*", "*iOS app's data artifacts*" and "*iOS app's RAM artifacts*".

5.3.3.1.1 Android app's data artifacts

Table 22 lists all the identified artifacts on Android OS. By examining the contents of the "*ezvizlog.db*" database an investigator can find information about the added CCTV system such as its serial number, WAN IP, and user's interaction with it. This information is stored within the "*content*" column of the "*event*" table in *JSON*-formatted key, value pairs.

Details about added CCTV system's active channels and their friendly names are saved within the "*channelinfo*" table of the "*database.hik*" database. The "*deviceinfo*" table of the same file holds information about the system's serial number, WAN IP, and user's "*Hik-connect*" account credentials although these fields were found most of the time encrypted with *AES*/*CBC* and encoded in *base64*.

All media files created through the app by the user are put under the *"/media/0/Pictures/*" directory in a folder named "*Hik-Connect Album*". Intelligence related to these files such as the originating camera can be retrieved from the "*images*" table of the "*image.db*" database.

If users log in to their "*Hik-Connect*" accounts or use the app in "*Visitor Mode*" several files are created under the app's "/*files*" directory and one file is created under the app's "/*shared_prefs*" directory. The latter is an XML file and its name consists of the "*user-ID*", a *32*-character long alphanumerical string which is distinctive per user. Each "*Hik-Connect*" account's "*user-ID*" is unique and used across both Android and iOS applications to identify that account.

By inspecting the "*user-ID.xml*" file one can recover when the user logged in to the app with that account as well as determine certain users' actions. The first *5*-character of "*user-ID*" also exist in the names of several files created under the *"/files*" directory.

Many of these files were encrypted realm databases. The most forensically valuable of them are two realm databases, "*devmgr.user-ID{5}.sec.realm*" and "*hc.realm*".

The first one was encrypted with a *64*-byte key whereas the second one was unencrypted. After using the app's RAM to recover the decryption key of the "*devmgr.user-ID{5}.sec.realm*" database investigators can view its contents. The type of information that lies within this database is similar to the "*ezvizlog.db*" database. However, unlike the "*ezvizlog.db*" this database also stores whether the CCTV system was accessed by its bind "*Hik-Connect*" or a shared account. To the same extent, this database does not store any user's interactions with the CCTV system. It's worth mentioning though that when the user has logged in with a "*Visitor Mode*" account this realm database is not populated with any of the aforementioned information and remains empty. Furthermore, if no account is used then "*user-ID.xml*", "*devmgr.user-ID{5}.sec.realm*" and the rest of the files under the "/*files*" directory does not get created at all.

Details about Wi-Fi networks that the mobile device was connected to while the application was used are populating the "*hc.realm*" database.

More interesting files can be found under the "*/shared_prefs*" directory. For example, the "*default.xml*" file stores whether an account is currently using the app or not and can also document certain users' actions when no account is used.

Furthermore, "*system_config.xml*" keeps track of the app's network traffic. Viewing this file can help determine the amount of data (mobile, Wi-Fi) consumed both daily and monthly during the app's usage.

The presence of the "*videoGo_device_info.xml*" file indicates that the user has enabled access to the CCTV system's "*Remote Configuration*" operations.

Additionally, the "*cache*" folder found under *"/media/0/Android/data/com.connect.enduser /okhttp*" directory can be proven useful. This folder contains cache files that are created while using the application. Even though their content varies and cannot be always verified, these files store information such as the user's name and email, the CCTV system's serial number, etc.

Last but not least, an investigator should also examine OS native files and third-party apps that store application's usage and media files' views (e.g., "*frosting.db*", "*usagestats*", "*Gallery*", etc.). These files are not included in Table 22 but still could provide insights regarding how often the application was used and which media files were viewed recently.

A more thorough interpretation of the most useful artifacts of Table 22 can be found in Appendix C.

Table 22 Identified artifacts on the Android OS at the HIKVISION mobile app research work

Artifact	Format	Information About
/databases/ezvizlog.db	SQLite	-CCTV system: (IP, S/N, etc.) -user's actions: (e.g., Live View)
/databases/database.hik	SQLite	-CCTV system's channels
/databases/image.db	SQLite	-user's created media through the app.
/files/devmgr.user-ID{5}.sec.realm	realm -Encrypted	-CCTV system: (IP, S/N, sharing status, etc.)
/files/hc.realm	realm	-connected WiFi networks while using the app.
/shared_prefs/user-ID.xml	XML	-user's login date -user's actions: (Live View, Playback)
/shared_prefs/default.xml	XML	-user's logon type -user's actions: (Live View, Playback)
/shared_prefs/videoGo_device_info.xml	XML	-exists if "Remote Configuration" is enabled
/shared_prefs/system_config.xml	XML	-network traffic of the app
/media/0/Pictures/Hik-Connect Album	folder	-media files stored through the app
/media/0/Android/data/com. connect.enduser/okhttp/cache	folder	-CCTV system: (IP, S/N, etc.) -user's account: (name, email, etc.)

5.3.3.1.2 Android app's RAM artifacts

RAM was an essential piece of evidence in order to decrypt "*devmgr.user-ID{5}.sec.realm*" database. Apart from collecting an application's RAM *fridump3* can also execute *strings* command against it. Opting for this option the command's result will be stored in an output text file. The examiner can evaluate this file in search of relative artifacts. In this research work, this option was chosen.

After scrutinizing the output file, the decryption key was spotted as can be seen in Figure 31. Utilizing *CyberChef* the *64*-character long decryption key was converted to its *128*-hex representation. This was the required format for *Realm Studio* to decrypt and view "*devmgr.user-ID{5}.sec.realm*" contents.

It should be noted that within the output file, decryption keys of the rest of the encrypted realm databases were found as well but those files were forensically uninteresting.

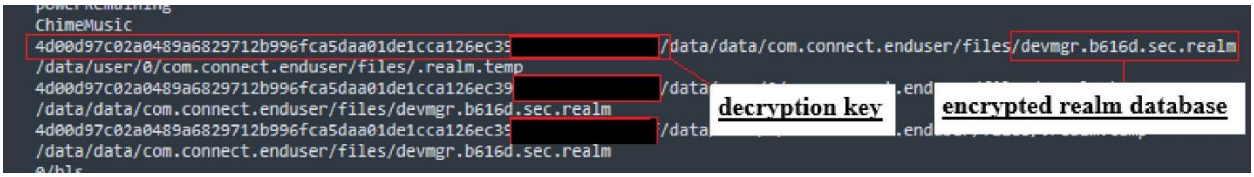

Figure 31 Fridump3 strings command's output file includes the decryption key of the HIKVISION mobile app's encrypted realm database

5.3.3.1.3 iOS app's data artifacts

Table 23 summarizes the discovered artifacts on iOS OS. The "*YSDCLogItem.sqlite*" database is the equivalent of Android's "*ezvizlog.db*". This database is slightly differently structured but stores the same information as "*ezvizlog.db*" within the "data" column of the "*YSDCLogItem*" table.

The "*database.hik*" database is present here as well and stores the same information.

"*TrafficStatistics.plist*" keeps track of the app's network traffic as "*system_config.xml*" does for Android. Evaluating this file can help determine the amount of data (mobile, Wi-Fi) consumed both daily and monthly during the app's usage.

If users log in to their "*Hik-Connect*" accounts or use the app in "*Visitor Mode*" a realm database gets created under the app's "*/Documents/EZ_REALM/*" directory. In this case, the database is unencrypted and its name consists of the "*user-ID*". By inspecting this file an investigator can retrieve similar artifacts to those of the "*devmgr.user-ID{5}.sec.realm*" database described earlier. Like before, when the user has logged in with a "*Visitor Mode*" account this realm database remains empty whereas if no account is used this database does not get created.

A *JSON* formatted text file that contains details about the CCTV system and the user's "*Hik-Connect*" account is the "*requestBase*" file that can be found under the *"/Documents*" directory.

All media files that have been created by the user can be found under the *"/Documents*" directory grouped in separate folders based on year, month, and day of their creation. App's user can also choose to download these media files to the *"/private/var/mobile/Media/DCIM/XXXAPPLE/*" directory. If the user opts for this option, all downloaded media files will be assigned to the "*Hik-Connect Album*" album.

Finally, an investigator should also examine OS native files that store the application's usage and media files' views (e.g., "*knowledgeC.db*", "*Photos.sqlite*", etc.). These files are not

included in Table 23 but still could provide insights regarding how often the application was used and which media files were viewed recently.

A more detailed interpretation of the most useful artifacts of Table 23 can be found in Appendix C.

Table 23 Identified artifacts on the iOS at the HIKVISION mobile app research work

Artifact	Format	Information About
/Documents/DCLOG/YSDCLogItem.sqlite	SQLite	-CCTV system: (IP, S/N, etc.) -user's actions: (e.g., Live View)
/Documents/database.hik	SQLite	-CCTV system's channels
/Documents/TrafficStatistics.plist	PLIST	-network traffic of the app
/Documents/EZ_REALM/user-ID.realm	realm	-CCTV system: (IP, S/N, sharing status, etc.)
/Documents/requestBase	text	-CCTV system: (IP, S/N, etc.) -user's account: (name, email, etc.)
/Documents/YYYY/MM/DD	folder	-user's created media through the app.
/private/var/mobile/Media/DCIM/XXXAPPLE/	folder	-user's created media through the app are assigned to "Hik-Connect Album".

5.3.3.1.4 iOS app's RAM artifacts

The iOS app's realm database was unencrypted contrary to the Android app's encrypted ones. This means that iOS App's RAM was not needed for viewing its contents. Therefore, RAM was not examined further. Nonetheless, investigators may uncover hidden artifacts during an app's RAM analysis that could be proven useful to an investigation such as the app's user's credentials, etc.

5.3.3.2 App's behavior

In this section, the way the application handles some of the aforementioned artifacts is described. This section was considered essential as after reviewing all collected evidence sources, certain peculiarities between artifacts were noted. An investigator should be aware of these details when dealing with this application. These variations are explained in the following sub-sections: "*Android app's characteristics*" and "*iOS app's characteristics*".

5.3.3.2.1 Android app's characteristics

If a user logs out from either a "*Hik-Connect*" or a "*Visitor Mode*" account, files related to that account will not be deleted. This includes both "*user-ID.xml*" and the realm databases under the *"/files*" directory. The encrypted realm databases can no longer be decrypted using the app's RAM though as the application no longer mounts them.

Nonetheless, this information can at least be used as an indication of how many accounts have historically been used in the app. Furthermore, when such an account is used many entries from the "*event*" table of the "*ezvizlog.db*" database is getting deleted. This hinders the complete recovery of the information stored within. On the other hand, when no account is used records of this database remain intact.

Moreover, if the application gets uninstalled user's created media files will also remain undeleted.

5.3.3.2.2 iOS app's characteristics

Similarly to the Android app, when a user logs out from either a "*Hik-Connect*" or a "*Visitor Mode*" account their realm database remains undeleted. Likewise, when such an account is used many entries from the "*YSDCLogItem*" table of the "*YSDCLogItem.sqlite*" database are getting deleted. These entries are left unaffected though if no account is used.

In case of uninstalling the application user's created media files get deleted, except for media files that have been downloaded to the "*/private/var/mobile/Media/DCIM/XXXAPPLE/*" directory.

5.3.3.3 Verifying user actions

Verifying the user's interaction with the CCTV system was a challenging task. Firstly, the following actions are allowed when a user accesses a CCTV system using the app:

- "***Live View***": view CCTV cameras' live footage.
- "***Playback***": view CCTV system's stored recordings.
- **Create media files**: the ability to either create screenshots or record videos from the CCTV system's footage.

Additional operations are available if the user has enabled the CCTV system's remote configuration:

- "***Basic Information***": view the CCTV system's basic information.
- "***Time Configuration***": sync the CCTV system's time with the mobile device's system time.
- "***Change Password***": modify the CCTV system's user password.

- "***Recording Schedule***": modify the CCTV system's recording schedule. This option allows a user to enable/disable the recording from a CCTV system's camera.
- "***Normal Event***": enable/disable the CCTV system's normal detection events (e.g., Motion Detection).
- "**Smart Event**": enable/disable the CCTV system's smart detection events (e.g., Intrusion Detection).

From the above-listed actions "*Live View*", "*Playback*" and the user's creation of media files could successfully be verified. The first two actions can be determined by inspecting the previously mentioned "*ezvizlog.db*" and "*YSDCLogItem.sqlite*" databases respectively. For instance, a user's "*Live View*" action created the following entry within the "*event*" table of the "*ezvizlog.db*" database (See Figure 32). The keys within the red boxes in Figure 32 indicate the following:

- "***serial***": accessed CCTV system's serial number.
- "***start_t***": when the user accessed the CCTV system's "*Live View*" footage. This timestamp is stored in the mobile device's local time.
- "***stop_t***": when the user exited the CCTV system's "*Live View*" footage. This timestamp is stored in mobile device's local time.
- "***via***": access type. The value of "*1*" denotes "*Live View*" access.
- "***systemName***": if this key's value is "*app_local _play*" then this entry is related to either "*Live View*" or "*Playback*" actions.

The same information would be stored if the user's action was "*Playback*". The only difference would be the "*via*" key's value which would be "*2*". The same entries would populate the "*YSDCLogItem.sqlite*" database as well.

All media files created by the user's actions can either be traced back to "*image.db*" or by examining the *"/Documents/YYYY/MM/DD*" directory.

Unfortunately, the user's actions regarding remote configuration did not leave any significant traces to the aforementioned databases and could not be verified successfully.

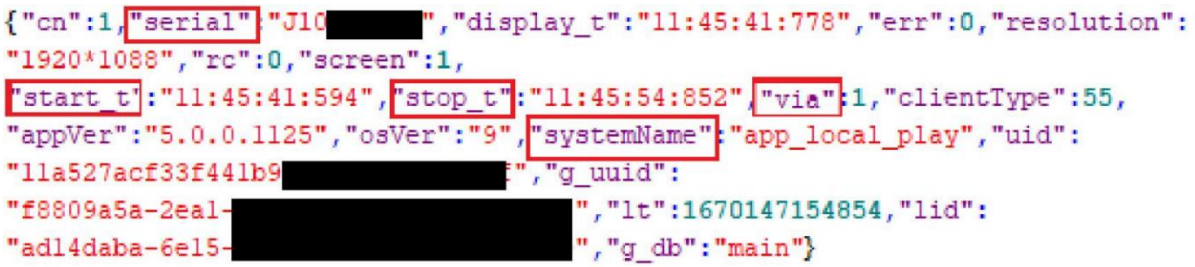

Figure 32 A user's "Live View" action created this entry within the "event" table of the HIKVISION mobile app "ezvizlog.db" database

5.3.3.4 Contributing to FOSS

Based on the results of this research work, code was contributed to the *ALEAPP* [134] and *iLEAPP* [135] software. In particular, SQLite queries were developed for recovering evidentiary data from "*ezvizlog.db*", "*image.db*", "*database.hik*", and "*YSDCLogItem .sqlite*" databases. These queries were incorporated into *Python* parsers that were used to enhance the capabilities of these tools. Both queries and parsers are available in *ALEAPP* and *iLEAPP* official repositories. An *ALEAPP* report of the parser's results is presented below (See Figure 33). This concludes the "*Results"* section of this research work.

5.4 Discussion

5.4.1 *HIKVISION* log records

Former studies with reference to the *HIKVISION* file system's log records were limited, yet these artifacts may be pivotal in certain investigations. Among other information, these log records document the user interaction with the CCTV system. Previously, this evidence source's availability relied on accessing the CCTV system's GUI/WebUI as commercial tools, including the manufacturer's native application, were not designed to analyze log records or failed to analyze them efficiently.

Using this research work as a point of reference, together with the utility which was developed as part of it, a forensic examiner can recover the stored log records from the *HIKVISION* file system, regardless of access to the GUI/WebUI. Exploiting this evidence source, they may be able to answer a number of important questions such as who gained access to the CCTV system and when, as well as which actions the logged-in user took. These questions may have remained unanswered up to this point.

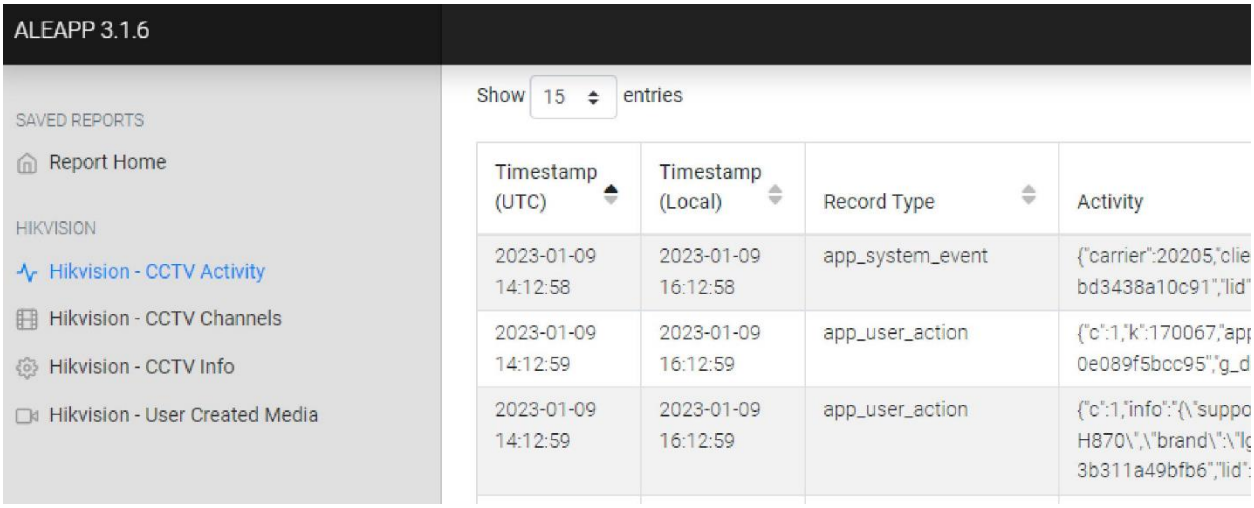

Figure 33 An ALEAPP report of HIKVISION parser's results

5.4.1.1 Limitations

Nevertheless, this research work has its limitations. Despite employing many CCTV systems to increase diverse log types' creation, many of them were never created in this research work. This is due to the equipment that was used. More expensive CCTV systems and cameras have more capabilities that could potentially generate different types of logs (e.g., *Object*/*Face Detection*). What is more, the identified log types were interpreted based on how the system's GUI/WebUI translated them. Either an update of the system's GUI/WebUI or a modification in a log type's hex representation could lead the *Hikvision Log Analyzer* to report inaccurate results.

Nonetheless, to deal with these challenges, *Hikvision Log Analyzer* provides the means to manually verify its results. Moreover, it is extensible as any volunteer who wishes to contribute to this project can assist with either mapping for undocumented log types or providing code for parsing currently unsupported log types. Its future maintenance will be partially affected by its contributors as there is currently no access to other log types.

5.4.2 *HIKVISION* mobile app

No previous studies with reference to the forensic analysis of a *HIKVISION* mobile application that can be used to remotely access CCTV systems were found yet the data stored within them can be proven useful in many investigations.

The methodology adopted provides the reader with insights related to the underlying technology and features of *HIKVISION* mobile application. Its dynamic approach generated a plethora of artifacts which resulted in a more efficient forensic evaluation of the app.

Among other things, the accumulated findings can help determine the user of the application, the IP of the CCTV system that was remotely accessed, and certain user actions (*Live View*/*Playback*/*Create Media Files*).

Additionally, these results demonstrate techniques and tools that tackle with modern challenges of digital forensics posed by realm databases' encryption. The analyst who utilizes them should be able to both decrypt and exploit their contents.

By referencing this research work in conjunction with *ALEAPP* and *iLEAPP* reports, an investigator should now be able to address various questions that previously might have gone unanswered.

5.4.2.1 Limitations

However, this research work comes with its own set of limitations. The accumulated findings from this research work might be partially or entirely altered based on updates to the analyzed mobile application. Also, using other types of equipment (e.g., *Pyronix* platform) may have generated more artifacts than those identified here.

5.5 Conclusion

CCTV surveillance systems are considered ubiquitous IoT solutions which more often than not become crime witnesses. Furthermore, these devices can be remotely accessed and configured using a plethora of applications. As a result, their digital forensic analysis is paramount to certain investigations.

HIKVISION is a company that manufactures such IoT devices, ships them with its proprietary *HIKVISION* file system and provides diverse applications that enable their remote control and configuration. Neither these applications nor the log records from the *HIKVISION* file system however have been forensically researched before leaving certain investigative questions unanswered. These answers depended on the exploitation of information that occasionally remained unidentified or uninterpreted within the log records of the *HIKVISION* proprietary file system and the most widely used *HIKVISION* mobile application.

In this chapter, the aforementioned sources of evidence are studied. Taking advantage of the insights shared here and using the developed *Hikvision Log Analyzer* application, extracting, parsing, interpreting, and evaluating available log records from the *HIKVISION* proprietary file system can be carried out effortlessly. Of course, this tool has some known limitations that can be resolved with the help of the digital forensic community. In addition, by referring to the findings presented here along with the code contributed to *ALEAPP* and *iLEAPP*, any valuable artifacts residing within the predominant *HIKVISION* mobile application can be extracted with ease. Hence, these resources provide the investigators with the means to address questions that previously could remain unanswered. Therefore, the goals outlined in **contribution iii** (See **Chapter 1** – **section 1.3**) have been successfully met.

Chapter 6: Digital Investigation of *AJAX Systems* SECSYS IoT Devices

6.1 Introduction

This chapter includes the forensic examination of the *AJAX Systems* SECSYS IoT products. *AJAX Systems* is a Ukrainian-based manufacturer of IoT security equipment. The company also offers applications available to multiple operating systems, that can allow configuration and monitoring of its appliances. According to its 2022 growth report [19], *AJAX Systems* has entered USA and Canada markets and is currently present in *169* countries in total. Despite the problems due to the war the company increased its revenue by *35*% and its end users surpassed *1,8* million worldwide.

6.1.1 Contribution of this chapter

This chapter responds to the objectives outlined in **contribution iv** (Refer to **Chapter 1** – **section 1.3**). Primary findings from this research work include:

- The exploration of some of the capabilities that the *AJAX Systems* mobile application offer to end users.
- The presentation of the artifacts that can be obtained from the forensic analysis of this mobile application and how they could be leveraged in real cases.
- Contributing to FOSS by developing *AJAX Systems Log Parser* [159], a new open-source utility that can be utilized to analyze some of the aforementioned artifacts.

6.1.2 How this chapter is organized

Section 6.2 details the research work regarding the digital investigation of the *AJAX Systems* mobile application that is used to remotely control *AJAX Systems* SECSYS products. Discussion about the identified findings is covered in **section 6.3**, and the chapter is concluded in **section 6.4**.

6.2 SECSYS

SECSYS devices can be configured using either desktop or mobile applications. In the event of an incident the information that gets stored within these applications can often be proven crucial to its investigation. Therefore, their forensic examination is needed as a mean to decipher the information they may collect.

AJAX Systems offers a mobile application for the end user of its products which is available for both Android and iOS operating systems. This mobile application is called "*Ajax Security System*" [160]. It is designed to allow the end users to both configure and monitor their IoT SECSYS.

This source of evidence is still unexplored by both commercial and open-source digital forensic software. Things could get worse in case of an incident (e.g., burglary, homicide) occurring at a place protected by *AJAX Systems* as important investigative questions such as when the incident happened, which IoT devices witnessed it and how was the SECSYS configured to operate may remain unanswered.

In the following section the equipment used and the methodology employed are introduced.

6.2.1 Equipment

Two fresh mobile devices and the *AJAX Systems* kit of IoT SECSYS devices named "*StarterKit*" were employed. This kit included a hub, a motion detector, an opening detector and a key fob with panic button. These IoT devices were installed into a laboratory. The mobile devices were a *LG G6* (*H870*) with Android 9 (security patch dated May 2019) and an *iPhone X* (*A1901*) with iOS 15.5. Root access was gained for both devices in order to obtain full file system access. The *LG* device was rooted with *Magisk* [124] whereas the *iPhone* was jailbreaked using *palera1n* [142].

The analysis of this research work was performed on a *Windows 10 Pro* (*21H2*) workstation. ADB was used for the majority of data exchange with the Android phone whereas *SSH* was mainly used with the iPhone. Even though these tools may not be considered as a forensically sound method to retrieve data from a piece of evidence and either commercial tools or more appropriate methods would most probably be used in a real investigation, this method offered the necessary versatility for the number of conducted experiments. At the end of the experiments an *FFS* image was acquired from both devices using *Magnet Acquire* [144] for the Android and *libimobiledevice* [145] for the iOS device. The purpose of these images was to locate any residual artifacts that could be missed if only *ADB* and *SSH* were preferred.

For the examination of application's data, *X-Ways Forensics* was utilized [126]. Additionally, as the application partially stored data in SQLite and realm database formats, *DB Browser for SQLite* [127] and *Realm Studio* [152] were used for viewing this information.

Frida [153], *fridump3* [154] and *CyberChef* [146] were also deployed to retrieve application's RAM and search for realm databases' decryption keys.

The hardware and software equipment used in this work is presented in Tables 24 and 25 respectively. *AJAX Systems* mobile application along with its versions examined in this research work are listed in Table 26.

Table 24 Hardware equipment used in the AJAX Systems mobile app research work

Hardware	Model–Version
AJAX Systems hub	Hub
AJAX Systems opening sensor	DoorProtect
AJAX Systems motion sensor	MotionProtect
AJAX Systems key fob	SpaceControl
LG G6	H870 - Android 9 (SPL May 2019)
iPhone X	A1901 – iOS 15.5
PC workstation	Windows 10 Pro (21H2)

Table 25 Software used in the AJAX Systems mobile app research work

Software	Version
Magisk	23
Palera1n	1.4.0
X-Ways Forensics	20.3 SR-4
ADB (Platform-Tools for Windows)	33.0.3
SSH	OpenSSH_for_Windows_8.1p1, LibreSSL 3.0.2
Magnet Acquire	2.59.0.32716
libimobiledevice	1.3.0
DB Browser for SQLite	3.12.2
Realm Studio	13.0.2
Frida	16.0.7
fridump3	-
CyberChef	9.55.0

Table 26 Versions of the AJAX Systems mobile app researched in this research work

Application	Version
AJAX Security System (com.ajaxsystems)	Android versions- 2.25.0, 2.25.1 and 2.25.2
AJAX Security System (systems.ajax.iosapp)	iOS versions - 2.22.1, 2.22.2, 2.22.3 and 2.22.4

6.2.2 Methodology

The methodology adopted consisted of three phases namely *Reconnaissance*, *Preparation/ Collection*, and *Analysis*.

6.2.2.1 Reconnaissance

At this phase, there was an introduction to some of the "*Ajax Security System*" mobile application's capabilities. This step was considered beneficial for the purpose of understanding the

application's features, its artifacts and some of the *AJAX Systems* technologies. The application is available at both Android's and iOS's official repositories (Play Store, App Store). Both Android and iOS application offered similar capabilities so one of them (Android) is analyzed.

To begin with, users can start using the application by logging in with their *AJAX Systems* accounts. If they do not have one, they can always create a new. After login two different things could happen. Firstly, if an existing IoT SECSYS is assigned to their accounts, logged in users would regain their access to it. Otherwise, the logged in user can create a new IoT SECSYS through the app by registering a hub along with any other device they want to add to it. A hub is regarded as the "*brain*" of such an IoT SECSYS. The account who created this new SECSYS is considered the *Admin* user of it. This type of user has full access to both IoT SECSYS and mobile application's settings. Some of the actions available to such a user type are listed below:

- **View/Edit App Settings**: The users can both view and edit account's information (name, email, etc.), other active app sessions (if account is also logged in other applications' instances), account's protection (enable/ disable app's two-factor authentication) and app's settings (e.g., enable/disable app's passcode lock protection).
- **Configure Devices**: The users can either add (pair) or remove (unpair) appliances from the IoT SECSYS. They can also configure IoT security system's paired devices, choosing individual settings for each appliance (device name, delay time, etc.).
- **Configure Hub**: As far as the hub of the SECSYS is concerned, this device offers extra options due to the nature of its usage and therefore is separated from the rest of the SECSYS appliances. For example, the users can configure the hub so as to share access to their SECSYS with other *AJAX Systems* accounts (can choose level of access for each shared account). Moreover, the users can create arming/disarming security schedules, set the hub to send cellular notifications (SMS/Calls) to the *AJAX Systems* accounts (provided that a SIM card has been inserted to it) and more.
- **Create/Delete Rooms**: The users can create and manage separate rooms within their IoT system as well as assign different devices to each room.
- **View Notifications**: The user can view notifications related to the IoT security system. The notifications can vary from events recorded from the SECSYS paired devices to events related to the SECSYS operation state and arming status.

- **Video Surveillance**: The users can integrate third-party (*HIKVISION*/*Safire*/*Uniview*) video surveillance equipment to their IoT SECSYS. This method permits access to video footage recorded from the video surveillance equipment through the application. This type could not be examined further as access to such a device was required.

As previously mentioned, if access to the IoT SECSYS is shared with other *AJAX Systems* accounts, their permissions can be adjusted by the SECSYS Admin user. This means that shared accounts could have as many available actions as the SECSYS *Admin* user type or their access could be restricted to only a small number of them. The communication between the *AJAX Systems* mobile app and the IoT SECSYS formed in this research work is presented in Figure 34.

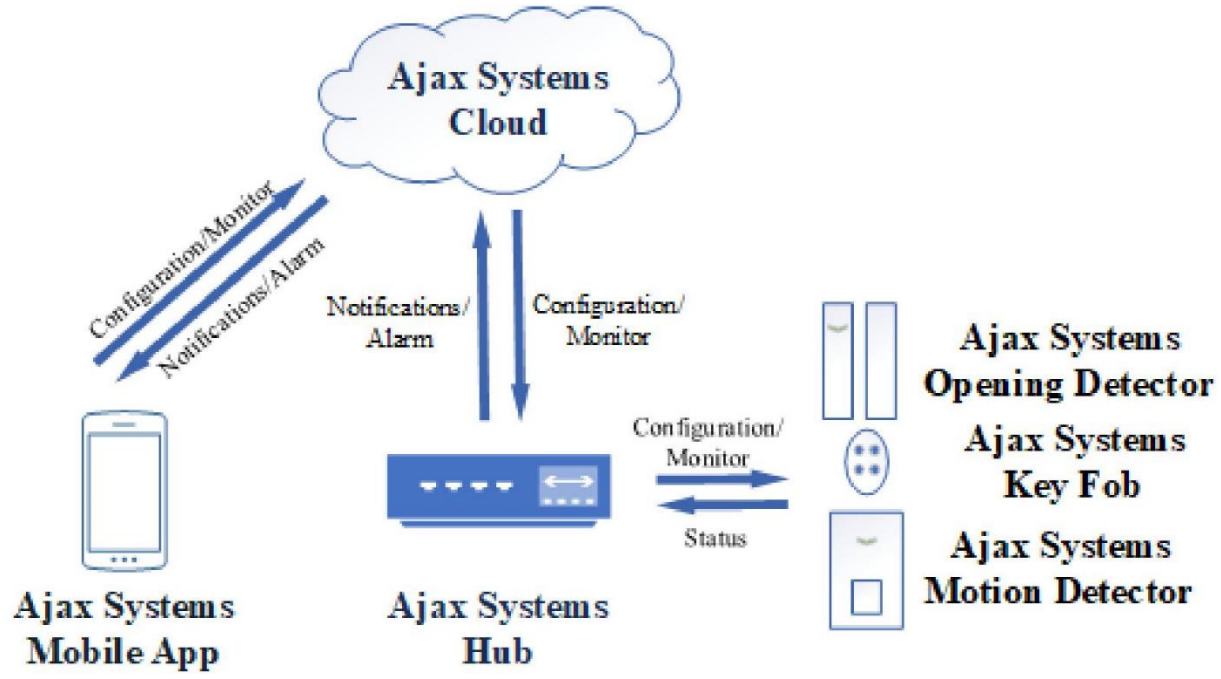

Figure 34 Information exchange between "AJAX Security System" mobile app and the IoT SECSYS of the AJAX Systems mobile app research work

After exploring the mobile application's capabilities, the authors begun the "*Preparation/Collection*" phase.

6.2.2.2 Preparation/Collection

During this phase the preparation steps of research were taken and the evidence to be examined was gathered.

To start with, the IoT SECSYS devices were installed into a laboratory. Afterwards, the application was installed on the mobile devices. A few *AJAX Systems* accounts were created for the purpose of the experiments. SIM cards were inserted to both the hub and the mobile devices in order to allow cellular notifications. Using the mobile application authors created a new IoT SECSYS, assigned the aforementioned appliances to it and configured them accordingly. The application was then used for a period of nearly two months in both mobile devices. During that period authors performed diverse actions using the application such as modifying both SECSYS and app's settings, viewing its notifications, sharing system's access with other accounts, configuring appliances and more. After the two-month period the collection phase started.

In order to identify how application's features affect forensic findings, a dynamic evidence collection process was followed. Application's data was collected using *ADB* and *SSH* commands in parallel with the experiments so as to be able to spot any variations in the artifacts and draw more solid conclusions from its forensic analysis. Furthermore, application's RAM was also collected using *Frida* and *fridump3* during the experiments in favor of decrypting application's realm databases whenever deemed necessary. Application's data and RAM were collected more than *70* times in total from both Android and iOS mobile devices. In the following Table (See Table 27) the actions performed prior to each evidence collection is demonstrated. At the end of the experiments an FFS image was acquired from both mobile devices in pursue of any residual artifacts outside the application's space.

Having gathered the above pieces of evidence the analysis phase started.

Table 27 Collected evidence per action performed in the AJAX Systems mobile app research work

Action Performed	No. of Android App's Data/RAM Evidence	No. of iOS App's Data/RAM Evidence
Install App	1 Data	1 Data
Login/Logout Admin Account	3 Data + 3 RAM	3 Data + 3 RAM
Configure Hub/ Share access/ Edit Permissions	3 Data + 3 RAM	3 Data
Login/Logout Shared Account	3 Data + 3 RAM	3 Data
View Notifications	4 Data + 4 RAM	4 Data
View/Edit App Settings	4 Data + 4 RAM	4 Data
Create/Delete Room	2 Data + 2 RAM	2 Data
Create/Delete Scenario-Security Schedule	2 Data + 2 RAM	2 Data
Configure Devices	3 Data + 3 RAM	3 Data
Uninstall App	1 Data	1 Data
Total	**26 Data + 24 RAM**	**26 Data + 3 RAM**

6.2.2.3 Analysis

This phase's primary objectives were to discover all valuable artifacts, determine how the application handles them and contribute to FOSS. The outcomes of the analysis phase are presented in the next section.

6.2.3 Results

6.2.3.1 Artifacts

Artifacts are divided in four sections based on the underlying OS and source of evidence: "*Android application's data artifacts*", "*Android application's RAM artifacts*", "*iOS application's data artifacts*" and "*iOS application's RAM artifacts*".

6.2.3.1.1 Android application's data artifacts

Table 28 lists all the potentially valuable artifacts on Android OS. Both "*ajax.realm*" and "*support.realm*" realm databases that reside within *"/files*" directory were encrypted with the same *64*-byte key. By exploiting the app's RAM their decryption key was successfully retrieved. The decryption key was the same among all collected evidence. The examination of their contents provides the investigator with vital information related to the IoT SECSYS under investigation and its users.

To start with, by exploring the tables of "*support.realm*" an investigator can find how many different users have historically logged in to their accounts through the app along with the chosen "*App Settings*" of the currently logged in user. The number of entries of "*AXAgreement*", "*AXConnection*" and "*AXKeepAlive*" tables indicate the number of users who have historically logged in through this app. "*AXLastLogin*" table stores the email ("*login*" column) of the account who last logged in through the app. If the user has chosen to protect app's usage with a *PIN* password, then this password can be found within the "*pin*" column of the "*AXLock*" table.

Following, "*ajax.realm*" is the "*goldmine*" of information related to the IoT SECSYS and its users. "*AXAccount*" table saves information about the account that is currently logged in (email, mobile, etc.) whereas "*AXUser*" table stores information about all registered users of this SECSYS (including users' permissions, mobiles, etc.). "*AXHub*" table saves details related to the hub of the SECSYS. These details include the geolocation ("*geo_gps_coords*" column) of the hub which could be proven pivotal if the exact location of the IoT SECSYS is needed. "*AXDevice*" table keeps track of the IoT devices that are part of the SECSYS along with their active settings. "*AXLogNew*"

table caches data related the notifications app's view. The number of this table's entries vary and depend on how many of these notifications have been retrieved from the *AJAX Systems* Cloud prior to collecting this evidence source. By default, this table stores the latest *20* notifications.

Subsequently, the application's programming interface (*API*) communication between the mobile application and *AJAX Systems* Cloud is stored in logfiles that can be found under *"/files/logs*" directory. Those logfiles are stored in separate folders named after the date they were created (in *DD-MM-YYYY* format). Unfortunately, only the latest logfile is available for immediate examination as all previous logfiles are stored in password-protected archives (.*zip* extension) within their corresponding folder. The password for these archives could not be uncovered. Logfiles are named using the current version of the app along with the Unix epoch time (in *UTC* offset) of their creation. An example filename of a logfile is "*v 2.25.1 (build #7843)-1674724363275.log*". Regardless of their availability, the intelligence that lies within these files can be fruitful to an investigation and was exploited while developing *Ajax Systems Log Parser*. For instance, these files store information related to the user (email, etc.), the IoT SECSYS (notifications, settings, etc.) and the app's communication ("*asking for passcode*", etc.).

Finally, the examiner should not omit the analysis of the mobile device's applications (native, third-party) that are chosen to receive SMS and calls. This is because if the hub is equipped with a SIM card and is configured accordingly it can notify users (if they are authorized to receive such notifications) in case of an alarm, event, malfunction etc. via either a call or a SMS directed to the registered mobile number of the *AJAX Systems* account.

Table 28 Identified artifacts on the Android OS at the AJAX Systems mobile app research work

Artifact	Format	Information About
/files/support.realm	realm-encrypted	-no. of historically logged in users -app's settings: (passcode, last logged in user, etc.)
/files/ajax.realm	realm-encrypted	-IoT security system: (logged in user's info, system's devices, notifications, hub's geolocation, registered users, etc.) -user/account: (email, mobile, etc.)
/files/logs/DD-MM-YYYY/*	folders containing .log and password protected .zip files	-API communication: (user's, system's and app's info)

6.2.3.1.2 Android application's RAM artifacts

RAM was an essential piece of evidence in order to decrypt "*ajax*.*realm*" and "*support*.*realm*" realm databases. Apart from collecting an application's RAM *fridump3* can also execute *strings* command against it. Opting for this option the command's result will be stored in an output text file. An examiner can evaluate this file in pursuance of relative artifacts. In this research work this

option was chosen. After scrutinizing the output file, the decryption key was spotted as can be seen in Figure 35. This key was common for both databases. Utilizing *CyberChef* this *64*-characters long decryption key was converted to its *128*-hex representation. This was the required format for *Realm Studio* to decrypt and view the aforementioned databases' contents.

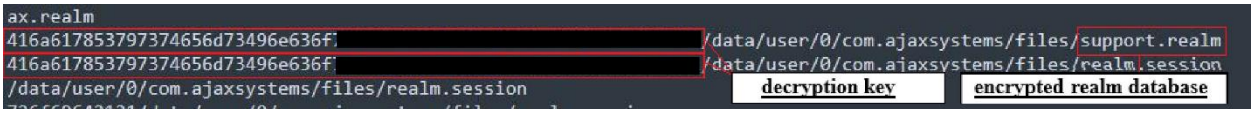

Figure 35 Fridump3 strings command's output file includes the decryption key of the AJAX Systems mobile app's encrypted realm databases

6.2.3.1.3 iOS application's data artifacts

Table 29 summarizes the discovered artifacts on iOS OS. The "*UserData.plist*" PLIST file located at *"/Documents/Settings*" directory, caches how many different users have historically logged in to their accounts through the app along with some of the chosen "*App Settings*" of the currently logged in user. If the user has chosen to protect app's usage with a password and enabled account's two-factor authentication, then the values of "*passcodeLock*" and "*2FAEnabled*" variables would be *True* respectively. The value of "*lastUsedLoginKey*" keeps track of the last logged in user's email. This file's contents seem similar to Android's "*support.realm*".

Next, the API communication between the mobile application and *AJAX Systems* Cloud is stored in logfiles that can be found under "*/Library/Caches/Logs*" directory. Those logfiles save similar information to those found on Android OS but contrary to them, they are all stored within the same folder and their contents are not archived nor password protected. These logfiles are named using the app's package name along with the date (in *UTC* offset) of their creation. An example filename of a logfile is "*systems.ajax.iosapp 2023-01-18--19-07-42-689.log*".

As explained before, the examiner should not skip the analysis of the mobile device's applications (native, third-party) that are chosen to receive SMS and calls.

Table 29 Identified artifacts on the iOS at the AJAX Systems mobile app research work

Artifact	Format	Information About
/Documents/Settings/UserData.plist	PLIST	-logged in user's info: (email,etc.) -app's settings:(passcode's status, last logged in user, etc.)
/Library/Caches/Logs/*	folders containing .log files	-API communication: (user's, system's and app's info)

6.2.3.1.4 iOS application's RAM artifacts

The iOS app did not make use of any kind of encrypted artifacts. This suggests that iOS App's RAM was not necessary for the analysis of the app. Therefore, RAM was not examined further. Nonetheless, investigators may uncover hidden artifacts during an app's RAM analysis that could be proven useful to an investigation such as app's user's credentials, etc.

6.2.3.2 Contributing to FOSS

Exploiting the findings of this research work, a piece of FOSS software was developed. In particular, this *Python* utility was designed to parse the aforementioned logs that store the *API* communication in order to recover any useful information related to either the IoT SECSYS (settings, notifications, etc.) or its users (email, permissions, etc.). This tool was named "*Ajax Systems Log Parser*" and is available on *GitHub* [159]. *Ajax Systems Log Parser* can analyze logs from both Android and iOS applications with the exception of those that reside within the password-protected archives. The parser reports its findings in easy-to-read *HTML* files. Figure 36 presents *Ajax Systems Log Parser* main application window and Figure 37 demonstrates an example report file of this tool. This marks the end of the "*Results*" section of this research work.

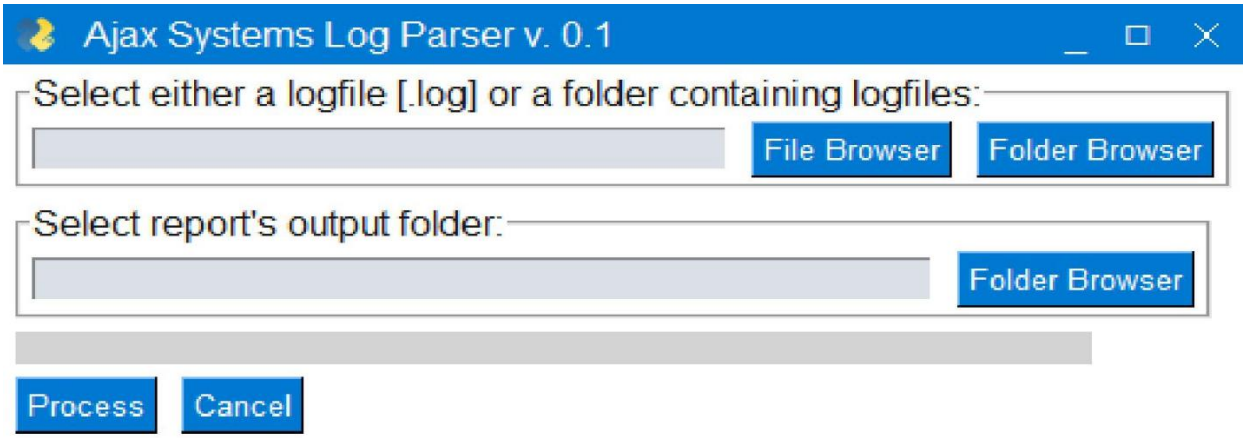

Figure 36 Main application window of the Ajax Systems Log Parser

Timestamp Raw	Device Name	Device Type	Device ID	Notification Type	Notification Event Type	Object ID	Room ID	Room Name
1674750727014	DoorWindowSensor	Door Sensor	10204439	Green	Closed	1404201717	2	Workshop
1674750676138	DoorWindowSensor	Door Sensor	10204439	Red	Opening detected	1404201716	2	Workshop

Figure 37 Part of Ajax Systems Log Parser's HTML report

6.3 Discussion

Previous studies with reference to the forensic analysis of IoT SECSYS devices and their companion apps are limited yet the information that gets stored within them can be proven vital in certain investigations. Moreover, no published research related to the examination of *AJAX Systems* applications was found.

Using this research work as point of reference along with *Ajax Systems Log Parser*'s results examiners can identify the structure of the IoT SECSYS under investigation, its registered users/devices, its stored notifications and its current configuration. Doing so they may be able to deal with a number of investigative questions that were unanswered up till now. For instance, in case of an incident (e.g., burglary) the recovery of the latest notifications can indicate when did the incident happened, which devices witnessed it and more. To the same extent, uncovering each registered device's configuration along with the system's API communication can help draw more firm conclusions related to that incident (disarming schedule deactivated the alarm, etc.).

6.3.1 Limitations

Nonetheless this research work has its own limitations. This work does not take into account the evidentiary data that could potentially reside within the IoT SECSYS devices themselves. For example, examining the hub's internal chip could provide supplementary information to the aforementioned. However, its analysis is a subject of future work. Furthermore, any updates in either the way the API communication operates or the way the mobile application stores this information, may cause Ajax Systems Log Parser to report inaccurate results.

6.4 Conslusion

SECSYS devices can be remotely configured and monitored by their end users through applications provided by their manufacturers. *AJAX Systems* is a company that produces such IoT devices and offers a variety of applications that permit their remote configuration and monitoring. These applications however have not been forensically researched before, leaving certain investigative questions unanswered.

In this chapter, this source of evidence is studied, in an attempt to fill a small part of the previously mentioned literature gap. To further assist investigators, a *Python* parser, namely the *Ajax Systems Log Parser* was developed. This tool can help parse some of the artifacts that can be found when dealing with this application. Hence, objectives detailed in **contribution iv** (See **Chapter 1** – **section 1.3**) have been achieved.

Chapter 7: Conclusion and future work

IoT has profoundly impacted our daily lives and promises to play an even more integral role in the future. The continued evolution of IoT devices will pave the way for smarter cities, enhancing many aspects such as healthcare, and transportation.

Given the rapid proliferation of IoT devices, it is inevitable that they are implicated in criminal activity, often serving as witnesses to crimes. As a consequence, the extraction and interpretation of the information they store becomes vital for both current and future investigations. This is a task for the specialized branch of digital forensics known as IoT Forensics.

During the course of this dissertation, a problem related to IoT Forensics was identified that was not addressed efficiently in the literature (See **Chapter 1** – **section 1.2** and **Chapter 2 – section 2.4**). This problem was that many CCTV and SECSYS IoT devices that are produced by leading IoT manufacturers and thus are more likely to appear at crime scenes, either have not been forensically examined at all, or have only been partially analyzed. Hence, some of the critical information they store remains unexplored in both the literature and by digital forensic software, resulting in unresolved investigative questions.

This dissertation set out to address the aforementioned problem. To achieve this, specific research objectives were established and these were met through the subsequent research contributions (See **Chapter 1** – **section 1.3**). As a result, new artifacts were identified and disseminated, novel open-source digital forensic software was developed, and existing software was updated based on the insights from the research conducted (See **Chapters 4** – **6**). The outcomes of this dissertation may serve as a useful resource for both investigators in this field and companies specializing in the development of digital forensic software.

7.1 Limitations

Before concluding, it is essential to acknowledge the limitations inherent in this dissertation. Recognizing these limitations provides a clearer context for interpreting and considering the accumulated findings. While these limitations have been individually discussed in their respective chapters (See **Chapter 3** – **section 3.5.1, Chapter 4** – **sections 4.4.1.1 – 4.4.2.1**, **Chapter 5** – **sections 5.4.1.1 – 5.4.2.1** and **Chapter 6** – **section 6.3.1**), this section provides a consolidated summary of the constraints encountered during the dissertation and underscores areas where caution should be taken in generalizing the results. The key limitations of this dissertation are outlined below:

- **Restricted evidence collection**: The collection of evidence was primarily limited to the mobile application used to control SECSYS IoT devices and the hard drive of the CCTV systems. Evidentiary data potentially residing within cloud infrastructure was not examined. Additionally, due to the absence of chip-off equipment, the internal memory chips of both SECSYS IoT devices and CCTV systems were not assessed. Their examination however is a topic of future work.
- **Firmware/mobile applications versions**: The findings of this dissertation could be partially or entirely affected by updates or upgrades to the mobile applications or the firmware of the CCTV systems (GUI/WebUI versions). Moreover, in such a scenario, the digital forensic software developed as part of this dissertation, as well as the tools to which code contributions were made, might yield inaccurate results.
- **Equipment-dependent Results**: The results of this dissertation are partly influenced by the specific equipment used. Employing SECSYS or CCTV IoT devices with different features could alter the outcomes. For example, CCTV systems and cameras with more advanced capabilities might generate other types of log records or handle evidentiary data differently.

To summarize, while this dissertation provides valuable insights and contributions to the field of IoT Forensics, it is important to consider its boundaries. The limitations highlighted above offer a comprehensive understanding of the scope of this dissertation and the constraints under which it operated.

In the following section potential topics of future work are shared.

7.2 Future Work

As the landscape of IoT Forensics continues to evolve, so do the opportunities for further exploration and refinement. Building upon the foundational research presented in this dissertation, there is a strong interest in accessing the evidentiary data stored within the employed IoT appliances. In pursuing this, additional artifacts beyond those already identified could be gathered and assessed in alignment with the broader context of this dissertation. In addition, the development of digital forensic software to retrieve evidentiary data from cloud resources presents another area of interest. A final topic of future work involves collecting these previously unexplored sources of evidence to simulate an "*Anti-forensics*" scenario, integrating evidence from all these sources.

References

[1] B. Hagler, "Council Post: How The Internet Of Things Is Transforming 21st-Century Manufacturing," Available: https://www.forbes.com/sites/forbestechcouncil/2020/04/17/how-the-internet-of-things-is-transforming-21st-century-manufacturing/. [Accessed 1 June 2023].

[2] M. Gava, "AI is the biggest technological breakthrough of the 21st century," 22 January 2022. Available: https://medium.com/geekculture/ai-is-the-biggest-technological-breakthrough-of-the-21st-century-6a6f78a823f8. [Accessed 1 June 2023].

[3] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues," *IEEE Communications Surveys Tutorials,* vol. 22, p. 1191–1221, 2020. doi:10.1109/COMST.2019.2962586.

[4] H. F. Atlam, E. El-Din Hemdan, A. Alenezi, M. O. Alassafi and G. B. Wills, "Internet of Things Forensics: A Review," *Internet of Things,* vol. 11, pp. 100-220, September 2020. doi: 10.1016/j.iot.2020.100220.

[5] N. Almolhis and M. Haney, "IoT Forensics Pitfalls for Privacy and a Model for Providing Safeguards," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2019. doi: 10.1109/CSCI49370.2019.00036.

[6] A. Alenezi, H. Atlam, R. Alsagri, M. Alassafi and G. Wills, "IoT Forensics: A State-of-the-Art Review, Challenges and Future Directions:," in *Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk*, Heraklion, 2019. doi: 10.5220/0007905401060115.

[7] A. MacDermott, T. Baker and Q. Shi, "Iot Forensics: Challenges for the Ioa Era," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Paris, 2018. doi: 10.1109/NTMS.2018.8328748.

[8] R. Lakshmanan, "FTC Slams Amazon with $30.8M Fine for Privacy Violations Involving Alexa and Ring," The Hacker News, 3 June 2023. Available: https://thehackernews.com/2023/06/ftc-slams-amazon-with-308m-fine-for.html. [Accessed 1 July 2023].

[9] H. A. Abdul-Ghani, D. Konstantas and M. Mahyoub, "A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Model," *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 3, 2018. doi: 10.14569/IJACSA.2018.090349.

[10] M. Msgna, "Anatomy of attacks on IoT systems: review of attacks, impacts and countermeasures," *Journal of Surveillance, Security and Safety,* vol. 3, no. 4, pp. 150-73, 2022. doi: 10.20517/jsss.2022.07.

[11] X. Zhang, O. Upton, N. L. Beebe and K.-K. R. Choo, "IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers," *Forensic Science International: Digital Investigation,* vol. 32, p. 300926, 2020. doi: 10.1016/j.fsidi.2020.300926.

[12] E. Köhler and D. Spiekermann, "Smart Home as a Silent Witness - A Survey," in *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, New York, NY, USA, 2022. doi: 10.1145/3528580.3528583.

[13] L. Urquhart, D. Miranda, L. Podoletz, J. Čas, P. De Hert, M. G. Porcedda and C. D. Raab, "Policing the smart home: The internet of things as 'invisible witnesses'1," *Information Polity,* vol. 27, no. 2, pp. 233-246, 2022. doi: 10.3233/IP-211541.

[14] R. Fahey, "App and fitness watch which helped convict Brit mum Caroline Crouch's killer," Mirror, 17 May 2022. Available: https://www.mirror.co.uk/news/world-news/how-app-fitness-watch-helped-26986593. [Accessed 1 June 2023].

[15] Statista, "IoT connected devices worldwide 2019-2030," Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. [Accessed 1 June 2023].

[16] C. S. Systems, "How Many CCTV Cameras in London? UK CCTV Numbers (Updated 2022)," Available: https://clarionuk.com/resources/how-many-cctv-cameras-are-in-london/. [Accessed 1 June 2022].

[17] "Closed-circuit television (CCTV)," College of Policing, 2021. Available: https://www.college.police.uk/research/crime-reduction-toolkit/cctv. [Accessed 1 June 2023].

[18] M. Jerian, "Essential Concepts and Principles for the Use of Video Evidence for Public Safety and Criminal Justice," 18 May 2023. Available: https://dl.ampedsoftware.com/video-evidence-principles.pdf. [Accessed 8 June 2023].

[19] "2022 Growth Report: Year of grit | Ajax Systems Blog," Available: https://ajax.systems/blog/2022-growth-report/. [Accessed 2 June 2023].

[20] E. Dragonas, C. Lambrinoudakis and M. Kotsis, "IoT Forensics: Analysis of a HIKVISION's mobile app," *Forensic Science International: Digital Investigation,* vol. 45, no. Supplement, p. 301560, 2023. doi: 10.1016/j.fsidi.2023.301560.

[21] E. Dragonas, C. Lambrinoudakis and M. Kotsis, "IoT Forensics: Investigating the Mobile App of Dahua Technology," *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, 2023, pp. 452-457, doi: 10.1109/CSR57506.2023.10224982.

[22] E. Dragonas, and C. Lambrinoudakis, "IoT Forensics: Analysis of Ajax Systems' mobile app for the end user," *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, 2023, pp. 446-451, doi: 10.1109/CSR57506.2023.10224992.

[23] Dragonas, E, Lambrinoudakis, C, Kotsis, M. IoT forensics: Exploiting unexplored log records from the HIKVISION file system. *J Forensic Sc*i. 2023; 00: 1–10. https://doi.org/10.1111/1556-4029.15349

[24] Dragonas, E, Lambrinoudakis, C, Kotsis, M. IoT forensics: Exploiting log records from the DAHUA technology CCTV systems. *J Forensic Sci*. 2023; 00: 1–14. https://doi.org/10.1111/1556-4029.15401

[25] E. Dragonas, "Forensic Analysis of Xiaomi IoT Ecosystem," SANS Digital Forensics and Incident Response Summit, 2021. Available: https://www.youtube.com/watch?v=4oVfHinPIz0&ab_channel=SANSDigitalForensicsandIncidentResponse. [Accessed 2 June 2023].

[26] E. Dragonas, "Forensic Analysis of Xiaomi IoT Ecosystem - DFRWS USA 2021," DFRWS USA 2021, 2021. Available: https://www.youtube.com/watch?v=zpCzctTUiWs&ab_channel=DFRWS. [Accessed 2 June 2023].

[27] E. Dragonas, " IoT Forensics: Exploiting an unexplored piece of evidence in CCTV investigations " SANS Digital Forensics and Incident Response Summit, Tokio, 2023. Available: https://www.youtube.com/@SANSForensics/playlists. [Accessed 2023].

[28] Z. Li and Z. Zuo, "Research on Electronic Data Recovery Technology of Dahua Embedded Video Surveillance System," *Forensic Science and Technology,* vol. 40, p. 445–449, 2015. doi: 10.16467/j.1008-3650.2015.06.004.

[29] J. Han, D. Jeong and S. Lee, "Analysis of the HIKVISION DVR File System," in *Digital Forensics and Cyber Crime*, vol. 157, J. I. James and F. Breitinger, Eds., Cham, Springer International Publishing, 2015, p. 189–199. doi: 10.1007/978-3-319-25512-5_13.

[30] S. Sandeepa, A. Reyaz and M. Silpa, "An Efficient Approach to Recover CCTV Video from Proprietary DVR File System," in *2018 International CET Conference on Control, Communication, and Computing (IC4)*, Thiruvananthapuram, 2018. doi: 10.1109/CETIC4.2018.8531073.

[31] R. Gomm, N.-A. Le-Khac, M. Scanlon and T. Kechadi, "An Analytical Approach to the Recovery of Data from 3rd Party Proprietary CCTV File Systems," 8 July 2016.

[32] L. Tobin, A. Shosha and P. Gladyshev, "Reverse engineering a CCTV system, a case study," *Digital Investigation,* vol. 11, p. 179–186, September 2014. doi: 10.1016/j.diin.2014.07.002.

[33] R. Gomm, R. Brooks, K.-K. R. Choo, N.-A. Le-Khac and K. Hew, "CCTV Forensics in the Big Data Era: Challenges and Approaches," 2020, p. 109–139. doi: 10.1007/978-3-030-47131-6_6.

[34] A. Ariffin, K.-K. R. Choo and Z. Yunos, "Forensic Readiness," in *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*, Elsevier, 2017, p. 147–162. doi: 10.1016/B978-0-12-805303-4.00010-1.

[35] Q. Lu, S. Shi, J. Xi, J. Zeng, Y. Li and X. Mao, "A method of time code retrieval for special format surveillance video based on file header comparison," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, 2018. doi: 10.1109/ISDFS.2018.8355348.

[36] A. Ariffin, J. Slay and K.-K. Choo, "Data Recovery from Proprietary Formatted Cctv Hard Disks," in *Advances in Digital Forensics IX*, vol. 410, G. Peterson and S. Shenoi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, p. 213–223. doi: 10.1007/978-3-642-41148-9_15.

[37] 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,* Vols. 42-43, p. 301451, October 2022. doi: 10.1016/j.fsidi.2022.301451.

[38] D. Giese, "Having fun with IoT: Reverse Engineering and Hacking of Xiaomi IoT Devices," Las Vegas, NV, USA.

[39] A. Awasthi, H. O. L. Read, K. Xynos and I. Sutherland, "Welcome pwn: Almond smart home hub forensics," *Digital Investigation,* vol. 26, p. S38–S46, July 2018. doi: 10.1016/j.diin.2018.04.014.

[40] SANS Digital Forensics and Incident Response, *Forensic Analysis of Apple HomePod & Apple HomeKit Environment w/ Mattia Epifani - SANS DFIR Summit,* 2020.

[41] S. Kim, M. Park, S. Lee and J. Kim, "Smart Home Forensics—Data Analysis of IoT Devices," *Electronics,* vol. 9, p. 1215, 28 July 2020. doi: 10.3390/electronics9081215.

[42] F. Servida and E. Casey, "IoT forensic challenges and opportunities for digital traces," *Digital Investigation,* vol. 28, p. S22–S29, April 2019. doi: 10.1016/j.diin.2019.01.012.

[43] S. Hutchinson and U. Karabiyik, "Forensic Analysis of the August Smart Device Ecosystem," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, Montreal, 2020. doi: 10.1109/ISNCC49221.2020.9297346.

[44] D. A. Orr and L. Sanchez, "Alexa, did you get that? Determining the evidentiary value of data stored by the Amazon® Echo," *Digital Investigation,* vol. 24, pp. 72-78, 2018.

[45] H. Chung, J. Park and S. Lee, "Digital forensic approaches for Amazon Alexa ecosystem," *Digital Investigation,* vol. 22, pp. S15-S25, 2017. doi: 10.1016/j.diin.2017.06.010.

[46] J. Hyde and B. Moran, "Alexa, are you Skynet? SANS Digital Forensicsand Incident Response Summit," 2017. Available: https://www.osdfcon.org/presentations/2017/Moran_Hyde-Alexa-are-you-skynet.pdf. [Accessed 29 January 2022].

[47] S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan and J. Cao, "IoT Forensics: Amazon Echo as a Use Case," *IEEE Internet of Things Journal,* vol. 6, no. 4, pp. 6487-6497, 2019. doi: 10.1109/JIOT.2019.2906946.

[48] H. Azhar and S. Bate, "Recovery of forensic artefacts from a smart home IoT ecosystem," in *CYBER 2019, The Fourth International Conference on Cyber-Technologies and Cyber-Systems*, IARIA, 2019, pp. 94-99.

[49] D. Pawlaszczyk, J. Friese and C. Hummert, ""Alexa, tell me …" - A forensic examination of the Amazon Echo Dot 3 rd Generation," *International Journal of Computer Sciences and Engineering,* vol. 7, no. 11, pp. 20-29, 2019. doi: 10.26438/ijcse/v7i11.2029.

[50] T. Olufohunsi, (2020). Alexa Forensics. doi: 10.13140/RG.2.2.25523.37928.

[51] M.-A. Youn, Y. Lim, K. Seo, H. Chung and S. Lee, "Forensic analysis for AI speaker with display Echo Show 2nd generation as a case study," *Forensic Science International: Digital Investigation,* vol. 38, 2021. doi: 10.1016/j.fsidi.2021.301130.

[52] S. Lorenz, S. Stinehour, A. Chennamaneni, A. B. Subhani and D. Torre, "IoT forensic analysis: A family of experiments with Amazon Echo devices," *Forensic Science International: Digital Investigation,* vol. 45, p. 301541, 1 June 2023. doi: 10.1016/j.fsidi.2023.301541.

[53] Y. Shin, H. Kim, S. Kim, D. Yoo, W. Jo and T. Shon, "Certificate Injection-Based Encrypted Traffic Forensics in AI Speaker Ecosystem," *Forensic Science International: Digital Investigation,* vol. 33, 2020. doi: 10.1016/j.fsidi.2020.301010.

[54] P. Moore, "SANS Digital Forensics and Incident Response - SANS DFIR Summit," 2018. Available: https://www.youtube.com/watch?v=dLQLJP0Cu7c. [Accessed 29 01 2020].

[55] M. Park and J. I. James, “Preliminary Study of a Google Home Mini,” J. Digit. Forensics, vol. 13, no. 3, pp. 163–174, 2019.

[56] G. Dorai, S. Houshmand and I. Baggili, "I Know What You Did Last Summer: Your Smart Home Internet of Things and Your iPhone Forensically Ratting You Out," in *In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018)*, 2018. doi: 10.1145/3230833.3232814.

[57] A. Akinbi and T. Berry, "Forensic Investigation of Google Assistant," *SN Computer Science,* vol. 1, no. 5, p. 272, 2020. doi: 10.1007/s42979-020-00285-x.

[58] S. Tristan, S. Sharma and R. Gonzalez, "Alexa/Google Home Forensics," 2020. doi: 10.1007/978-3-030-23547-5_7.

[59] S. Engelhardt, "Smart Speaker Forensics," *Business/Business Administration,* vol. 56, 2019.

[60] I. Yildirim, G. E. Bostanci and M. Guzel, Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and Google Assistant Build-In Smart Home Speakers, 2019. doi: 10.1109/UBMK.2019.8907007.

[61] H. Barral, G.-A. Jaloyan, F. Thomas-Brans, M. Regnery, R. Géraud-Stewart, T. Heckmann, T. Souvignet and D. Naccache, "A forensic analysis of the Google Home: repairing compressed data without error correction," *Forensic Science International: Digital Investigation,* Vols. 42-43, p. 301437, 1 October 2022. doi: 10.1016/j.fsidi.2022.301437.

[62] M. Epifani, "A journey into IoT Forensics - Episode 5 - Analysis of the Apple HomePod and the Apple Home Kit Environment (aka thanks RN Team!)," 2021. Available: https://blog.digital-forensics.it/2021/01/a-journey-into-iot-forensics-episode-5.html. [Accessed 29 January 2022].

[63] W. Jo, Y. Shin, H. Kim, D. Yoo, D. Kim, C. Kang, J. Jin, J. Oh, B. Na and T. Shon, "Digital Forensic Practices and Methodologies for AI Speaker Ecosystems," *Digital Investigation,* vol. 29, pp. S80-S93, 2019. doi: 10.1016/j.diin.2019.04.013.

[64] L. Lin, X. Liu, X. Fu, B. Luo, X. Du and M. Guizani, A Non-Intrusive Method for Smart Speaker Forensics, 2021. doi: 10.1109/ICC42927.2021.9500679.

[65] L. Dawson and A. Akinbi, "Challenges and opportunities for wearable IoT forensics: TomTom Spark 3 as a case study," *Forensic Science International: Reports,* vol. 3, p. 100198, July 2021. doi: 10.1016/j.fsir.2021.100198.

[66] S. Kang, S. Kim and J. Kim, "Forensic analysis for IoT fitness trackers and its application," *Peer-to-Peer Networking and Applications,* vol. 13, p. 564–573, March 2020. doi: 10.1007/s12083-018-0708-3.

[67] Y. H. Yoon and U. Karabiyik, "Forensic Analysis of Fitbit Versa 2 Data on Android," *Electronics,* vol. 9, p. 1431, 2 September 2020. doi: 10.3390/electronics9091431.

[68] F. Hantke and A. Dewald, "How can data from fitness trackers be obtained and analyzed with a forensic approach?," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Genoa, 2020. doi: 10.1109/EuroSPW51379.2020.00073.

[69] 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*, Toulouse, 2015. doi: 10.1109/ARES.2015.39.

[70] S. Becirovic and S. Mrdovic, "Manual IoT Forensics of a Samsung Gear S3 Frontier Smartwatch," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, 2019. doi: 10.23919/SOFTCOM.2019.8903845.

[71] O. A. Kehinde, *DIGITAL FORENSIC ANALYSIS OF SMART WATCHES,* TALLINN, 2020.

[72] D. Quick and K.-K. R. Choo, "IoT Device Forensics and Data Reduction," *IEEE Access,* vol. 6, p. 47566–47574, 2018. doi: 10.1109/ACCESS.2018.2867466.

[73] A. Boztas, A. R. J. Riethoven and M. Roeloffs, "Smart TV forensics: Digital traces on televisions," *Digital Investigation,* vol. 12, p. S72–S80, March 2015. doi: 10.1016/j.diin.2015.01.012.

[74] M. Epifani, "A journey into IoT Forensics - Episode 2 - Analysis of an LG Television (aka thanks VTO Labs for sharing!)," 21 December 2020. Available: https://blog.digital-forensics.it/2020/12/a-journey-into-iot-forensics-episode-2.html. [Accessed 8 June 2023].

[75] M. Epifani, "A journey into IoT Forensics - Episode 3 - Analysis of an Ematic Android TV OS Box (aka thanks VTO Labs for sharing!)," 30 December 2020. Available: https://blog.digital-forensics.it/2020/12/a-journey-into-iot-forensics-episode-3.html. [Accessed 8 June 2023].

[76] M. Hadgkiss, S. Morris and S. Paget, "Sifting through the ashes: Amazon Fire TV stick acquisition and analysis," *Digital Investigation,* vol. 28, p. 112–118, March 2019. doi: 10.1016/j.diin.2019.01.003.

[77] H. Zhou, L. Deng, W. Xu, W. Yu, J. Dehlinger and S. Chakraborty, "Towards Internet of Things (IoT) Forensics Analysis on Intelligent Robot Vacuum Systems," in *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2022. doi: 10.1109/SERA54885.2022.9806735.

[78] M. Epifani, "A journey into IoT Forensics - Episode 4 - Analysis of an iRobot Roomba 690 (aka thanks VTO Labs for sharing!)," 30 December 2020. Available: https://blog.digital-forensics.it/2020/12/a-journey-into-iot-forensics-episode-4.html. [Accessed 8 June 2023].

[79] M. Epifani, "A journey into IoT Forensics - Episode 1 - Analysis of a Samsung Refrigerator (aka thanks VTO Labs for sharing!)," 19 December 2020. Available: https://blog.digital-forensics.it/2020/12/a-journey-into-iot-forensics-episode-1.html. [Accessed 8 June 2023].

[80] J. Winkelman, K.-K. R. Choo and N.-A. Le-Khac, "IoT Database Forensics—A Case Study with Video Door Bell Analysis," in *A Practical Hands-on Approach to Database Forensics*, N. Le-Khac and K. R. Choo, Eds., Cham, Springer International Publishing, 2022, p. 233–249. doi: 10.1007/978-3-031-16127-8_7.

[81] S. Hutchinson, Y. H. Yoon, N. Shantaram and U. Karabiyik, "Internet of Things Forensics in Smart Homes: Design, Implementation, and Analysis of Smart Home Laboratory," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*, Virtual On line, 2020. doi: 10.18260/1-2--34868.

[82] N.-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens and K.-K. R. Choo, "Smart vehicle forensics: Challenges and case study," *Future Generation Computer Systems,* vol. 109, p. 500–510, August 2020. doi: 10.1016/j.future.2018.05.081.

[83] C. Meffert, D. Clark, I. Baggili and F. Breitinger, "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, 2017. doi: 10.1145/3098954.3104053.

[84] 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), New York, NY, USA, 2018. doi: 10.1109/TrustCom/BigDataSE.2018.00201.

[85] 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)*, Vienna, 2016. doi: 10.1109/FiCloud.2016.57.

[86] 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)*, Xi'an, 2018. doi: 10.1109/SmartIoT.2018.00-19.

[87] L. Babun, A. K. Sikder, A. Acar and A. S. Uluagac, "IoTDots: A Digital Forensics Framework for Smart Environments," *arXiv:1809.00745 [cs],* 3 September 2018. doi: 10.48550/arXiv.1809.00745.

[88] M. Hassan, G. Samara and M. Fadda, "IoT Forensic Frameworks (DFIF, IoTDOTS, FSAIoT): A Comprehensive Study," *International Journal of Advances in Soft Computing and its Applications,* vol. 14, p. 73–83, 20 April 2022. doi: 10.15849/IJASCA.220328.06.

[89] N. Koroniotis, N. Moustafa and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems,* vol. 110, p. 91–106, 1 September 2020. doi: 10.1016/j.future.2020.03.042.

[90] P. M. Shakeel, S. Baskar, H. Fouad, G. Manogaran, V. Saravanan and C. E. Montenegro-Marin, "Internet of things forensic data analysis using machine learning to identify roots of data scavenging," *Future Generation Computer Systems,* vol. 115, p. 756–768, February 2021. doi: 10.1016/j.future.2020.10.001.

[91] G. Surange and P. Khatri, "Integrated intelligent IOT forensic framework for data acquisition through open-source tools," *International Journal of Information Technology,* vol. 14, p. 3011–3018, 1 October 2022. doi: 10.1007/s41870-022-01025-5.

[92] R. Jacob and A. Nisbet, "A forensic investigation framework for Internet of Things monitoring," *Forensic Science International: Digital Investigation,* Vols. 42-43, p. 301482, October 2022. doi: 10.1016/j.fsidi.2022.301482.

[93] E. Oriwoh, D. Jazani, G. Epiphaniou and P. Sant, "Internet of Things Forensics: Challenges and Approaches," in *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Austin, 2013. doi: 10.4108/icst.collaboratecom.2013.254159.

[94] S. Zawoad and R. Hasan, "FAIoT: Towards Building a Forensics Aware Eco System for the Internet of Things," in *2015 IEEE International Conference on Services Computing*, New York City, NY, USA, 2015. doi: 10.1109/SCC.2015.46.

[95] 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)*, Sierre, 2015. doi: 10.1109/ICDIPC.2015.7323000.

[96] 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)*, Tirgu, 2017. doi: 10.1109/ISDFS.2017.7916508.

[97] M. Qatawneh, W. Almobaideen, M. Khanafseh and I. A. Qatawneh,. (2019). DFIM: A NEW DIGITAL FORENSICS INVESTIGATION MODEL FOR INTERNET OF THINGS. 24.

[98] A. Nieto, R. Rios and J. Lopez, "IoT-Forensics Meets Privacy: Towards Cooperative Digital Investigations," *Sensors,* vol. 18, p. 492, 7 February 2018. doi: 10.3390/s18020492.

[99] N. Scheidt and M. Adda, "Identification of IoT Devices for Forensic Investigation," in *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, Varna, 2020. doi: 10.1109/IS48319.2020.9200150.

[100] A. Hilgenberg, T. Q. Duong, N.-A. Le-Khac and K.-K. R. Choo, "Digital Forensic Investigation of Internet of Thing Devices: A Proposed Model and Case Studies," in *Cyber and Digital Forensic Investigations: A Law Enforcement Practitioner's Perspective*, N. Le-Khac and K. R. Choo, Eds., Cham, Springer International Publishing, 2020, p. 31–49. doi: 10.1007/978-3-030-47131-6_3.

[101] P. Agbedanu and A. D. Jurcut, "BLOFF: A Blockchain-Based Forensic Model in IoT," in *Advances in Information Security, Privacy, and Ethics*, S. Singh and A. D. Jurcut, Eds., IGI Global, 2021, p. 59–73. doi: 10.4018/978-1-7998-7589-5.ch003.

[102] J. Kim, J. Park and S. Lee, "An improved IoT forensic model to identify interconnectivity between things," *Forensic Science International: Digital Investigation,* vol. 44, p. 301499, March 2023. doi: 10.1016/j.fsidi.2022.301499.

[103] A. Akinbi, Á. MacDermott and A. M. Ismael, "A systematic literature review of blockchain-based Internet of Things (IoT) forensic investigation process models," *Forensic Science International: Digital Investigation,* Vols. 42-43, p. 301470, October 2022. doi: 10.1016/j.fsidi.2022.301470.

[104] SWGDE, *SWGDE Best Practices for On-Scene Identification, Seizure, and Preservation of Internet of Things (IoT) Devices (22-F-001-1.0).pdf,* 2022.

[105] SWGDE, *SWGDE Technical Notes on Internet of Things Devices_{v}{1}{.}0.pdf,* 2020.

[106] S. Alabdulsalam, K. Schaefer, T. Kechadi and N.-A. Le-Khac, "Internet of Things Forensics – Challenges and a Case Study," in *Advances in Digital Forensics XIV*, vol. 532, G. Peterson and S. Shenoi, Eds., Cham, Springer International Publishing, 2018, p. 35–48. doi: 10.1007/978-3-319-99277-8_3.

[107] Z. A. Baig, P. Szewczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, K. Sansurooah, N. Syed and M. Peacock, "Future challenges for smart cities: Cyber-security and digital forensics," *Digital Investigation,* vol. 22, p. 3–13, September 2017. doi: 10.1016/j.diin.2017.06.015.

[108] J.-P. A. Yaacoub, H. N. Noura, O. Salman and A. Chehab, "Advanced digital forensics and anti-digital forensics for IoT systems: Techniques, limitations and recommendations," *Internet of Things,* vol. 19, p. 100544, 1 August 2022. doi: 10.1016/j.iot.2022.100544.

[109] F. Bouchaud, G. Grimaud, T. Vantroys and P. Buret, "Digital Investigation of IoT Devices in the Criminal Scene," *Journal of Universal Computer Science,* vol. 25, p. 1199–1218, 2019. hal-02432740.

[110] M. Conti, A. Dehghantanha, K. Franke and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems,* vol. 78, p. 544–546, January 2018. doi: 10.1016/j.future.2017.07.060.

[111] R. C. Hegarty, D. J. Lamb and A. Attwood, "Digital Evidence Challenges in the Internet of Things," 2014.

[112] Y. C. Tok, C. Wang and S. Chattopadhyay, "Stitcher: Correlating digital forensic evidence on internet-of-things devices," *Forensic Science International: Digital Investigation,* vol. 35, p. 301071, December 2020. doi: 10.1016/j.fsidi.2020.301071.

[113] N. Zulkipli, A. Alenezi and G. B. Wills, "IoT Forensic: Bridging the Challenges in Digital Forensic and the Internet of Things:," in *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security*, Porto, 2017. doi: 10.5220/0006308703150324.

[114] U. Karabiyik and K. Akkaya, *Digital Forensics for IoT and WSNs,* arXiv, 2018.

[115] Á. MacDermott, T. Baker, P. Buck, F. Iqbal and Q. Shi, "The Internet of Things: Challenges and Considerations for Cybercrime Investigations and Digital Forensics:," *International Journal of Digital Crime and Forensics,* vol. 12, p. 1–13, January 2020. doi: 10.4018/IJDCF.2020010101.

[116] R. Montasari and R. Hill, "Next-Generation Digital Forensics: Challenges and Future Paradigms," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, 2019. doi: 10.1109/ICGS3.2019.8688020.

[117] T. Wu, F. Breitinger and I. Baggili, "IoT Ignorance is Digital Forensics Research Bliss: A Survey to Understand IoT Forensics Definitions, Challenges and Future Research Directions," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, Canterbury CA United Kingdom, 2019. doi: 10.1145/3339252.3340504.

[118] W. Yang, M. N. Johnstone, L. F. Sikos and S. Wang, "Security and Forensics in the Internet of Things: Research Advances and Challenges," in *2020 Workshop on Emerging Technologies for Security in IoT (ETSecIoT)*, Sydney, 2020. doi: 10.1109/ETSecIoT50046.2020.00007.

[119] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems,* vol. 92, p. 265–275, March 2019. doi: 10.1016/j.future.2018.09.058.

[120] I. Yaqoob, E. Ahmed, M. H. U. Rehman, A. I. A. Ahmed, M. A. Al-garadi, M. Imran and M. Guizani, "The rise of ransomware and emerging security challenges in the Internet of Things," *Computer Networks,* vol. 129, p. 444–458, December 2017. doi: 10.1016/j.comnet.2017.09.003.

[121] P. Lutta, M. Sedky and M. Hassan, P. Lutta, M. Sedky and M. Hassan, "The Forensic Swing of Things: The Current Legal and Technical Challenges of IoT Forensics," vol. 14, 2020.

[122] "Number of Xiaomi IoT connected devices 2018-2022," Available: https://www.statista.com/statistics/967485/worldwide-xiaomi-number-of-connected-devices/. [Accessed 9 June 2023].

[123] Research and Markets ltd., "Global Surveillance Camera Market: Analysis By System Type (Analog, IP Commercial, IP Consumer & Other Surveillance Camera), By Technology (Image Signal Processor, Vision Processor, Vision Processor + AI) By Region Size and Trends with Impact of COVID-19 and Forecast up to 2027," September 2022. Available: https://www.researchandmarkets.com/report/surveillance-camera. [Accessed 18 May 2023].

[124] "Download Magisk Manager Latest Version 25.2 For Android 2022," 4 June 2017. Available: https://magiskmanager.com/. [Accessed 4 January 2023].

[125] "Command-line tools | Android Studio | Android Developers," Available: https://developer.android.com/tools. [Accessed 4 July 2023].

[126] "X-Ways Forensics: Integrated Computer Forensics Software," Available: https://www.x-ways.net/forensics/index-m.html. [Accessed 4 January 2023].

[127] "DB Browser for SQLite," Available: https://sqlitebrowser.org/. [Accessed 9 June 2021].

[128] H. Don, "Notepad++," Available: https://notepad-plus-plus.org/. [Accessed 2 July 2023].

[129] "Overview - Dahua Technology," Available: https://www.dahuasecurity.com/uk/aboutus/introduction/0. [Accessed 11 August 2023].

[130] "Magnet DVR EXAMINER," Available: https://www.magnetforensics.com/products/magnet-dvr-examiner/. [Accessed 1 May 2023].

[131] "Video Investigation Portable 2.0 | Video Forensics Analysis," Available: https://www.salvationdata.com/business-list-page/video-investigation-portable/. [Accessed 1 May 2023].

[132] "HX-Recovery for DVR - CCTV Recovery, DVR Data Recovery, Security Camera Hard Drive Recovery Experts," Available: http://www.hxdvr.com/. [Accessed 1 May 2023].

[133] "Dahua Wiki," Available: https://dahuawiki.com/Software/Dahua_Toolbox/_Disk_Manager. [Accessed 5 August 2023].

[134] A. Brignoni, "ALEAPP," 2 January 2023. Available: https://github.com/abrignoni/ALEAPP. [Accessed 4 January 2023].

[135] A. Brignoni, "iLEAPP," 2 January 2023. Available: https://github.com/abrignoni/iLEAPP. [Accessed 4 January 2023].

[136] "ThisPersonDoesNotExist - Random AI Generated Photos of Fake Persons," Available: https://this-person-does-not-exist.com/en. [Accessed 11 May 2023].

[137] "FTK Imager," Available: https://www.exterro.com/ftk-imager. [Accessed 19 December 2022].

[138] D. Pawlaszczyk, "FQLite - Forensic SQLite Data Recovery Tool," 25 April 2023. Available: https://github.com/pawlaszczyk/fqlite. [Accessed 13 May 2023].

[139] bp2008, "DahuaLoginBypass," 2 April 2023. Available: https://github.com/bp2008/DahuaLoginBypass. [Accessed 8 May 2023].

[140] "OSFClone - Open source utility to create and clone forensic disk images," Available: https://www.osforensics.com/tools/create-disk-images.html. [Accessed 3 May 2023].

[141] "Arsenal Recon," Available: https://arsenalrecon.com/products/arsenal-image-mounter. [Accessed 14 May 2023].

[142] "palera1n," Available: https://palera.in. [Accessed 4 January 2023].

[143] "DMSS," Available: https://software.dahuasecurity.com/en/dmss. [Accessed 18 February 2023].

[144] "Magnet ACQUIRE," Available: https://www.magnetforensics.com/resources/magnet-acquire/. [Accessed 11 January 2023].

[145] "libimobiledevice · A cross-platform FOSS library written in C to communicate with iOS devices natively.," Available: https://libimobiledevice.org/. [Accessed 11 January 2023].

[146] "CyberChef," Available: https://gchq.github.io/CyberChef/. [Accessed 11 January 2023].

[147] I. Whiffin, "Mushy," Available: https://www.doubleblak.com/software.php?id=2.

[148] "HIKVISION UK PORTAL," Available: http://www.hikvisioneurope.com/uk/portal/?dir=portal/Software/Software%20Tools/Local%20Playback/V3.0.1.2. [Accessed 19 December 2022].

[149] E. Dragonas, "HikvisionLogAnalyzer/README.md at main · theAtropos4n6/HikvisionLogAnalyzer," Available: https://github.com/theAtropos4n6/HikvisionLogAnalyzer. [Accessed 3 May 2023].

[150] "HxD - Freeware Hex Editor and Disk Editor | mh-nexus," Available: https://mh-nexus.de/en/hxd/. [Accessed 19 December 2022].

[151] "Hikvision App Store," Available: https://appstore.hikvision.com/. [Accessed 4 January 2023].

[152] "Realm Studio: open, edit, and manage your Realm data," Available: https://www.mongodb.com/docs/realm-legacy/products/realm-studio.html. [Accessed 4 January 2023].

[153] "Frida • A world-class dynamic instrumentation toolkit," Available: https://frida.re/. [Accessed 4 January 2023].

[154] R. Paul, "fridump3," 3 January 2023. Available: https://github.com/rootbsd/fridump3. [Accessed 4 January 2023].

[155] "Hik-Connect - for End User - Apps on Google Play," Available: https://play.google.com/store/apps/details?id=com.connect.enduser&hl=en. [Accessed 28 March 2023].

[156] HIKVISION, "Hik-Connect," Available: https://www.hik-connect.com/views/login/index.html?returnUrl=http://www.hik-connect.com/devices/page&r=7982140459941564219&host=www.hik-connect.com&from=c17392dc2e6c405a931b#/. [Accessed 4 January 2023].

[157] "Security Systems Manufacturer for Home & Business," Available: https://www.pyronix.com/. [Accessed 4 January 2023].

[158] "My Dyn Account," Available: https://account.dyn.com/. [Accessed 11 January 2023].

[159] E. Dragonas, "theAtropos4n6/AjaxSystemsLogParser: A Python utility that can parse Ajax Systems mobile application's logs (Android/iOS).," Available: https://github.com/theAtropos4n6/AjaxSystemsLogParser. [Accessed 10 April 2023].

[160] "Ajax Security System - Apps on Google Play," Available: https://play.google.com/store/apps/details?id=com.ajaxsystems&hl=en&gl=US. [Accessed 28 January 2023].

APPENDIX A – Interpretation of the log records identified in the DAHUA Technology CCTV research work

Action performed	Log entries within "{Serial Number}_log.db" and "carved.db"		Log entry within internal memory/ exported text files	
	"modename" column	"optname" column	Type field	Event Type /Action fields
Login through the GUI/ WebUI.	Log_Ugm	Log_Login	ACCOUNT→ User logged in.	-
	Log_Ugm	Log_NetLogin	ACCOUNT→ User logged in.	-
Logout.	Log_Ugm	Log_Logout	ACCOUNT→ User Logout	-
Illegal Login (wrong credentials).	Log_Ugm	Log_UserUnauth	Alarm Type→ Illegal Login	Event Type:Illegal Login Event Action:Event Start
Add a new user.	Log_Ugm	Log_AddUser	ACCOUNT→ Add User	-
Delete an existing user.	Log_Ugm	Log_DelUser	ACCOUNT→ Delete User	-
Modify user's permissions.	Log_Ugm	Log_ModUser	ACCOUNT→ Modify User	-
Start - Physical tampering of a video channel (e.g. physically disconnect camera's cable)	Log_Event	LossStart	Alarm Type→ Video Loss	Event Type:Video Loss Event Action:Event Start
End - Physical tampering of a video channel (e.g. re-connect camera's cable)	Log_Event	LossEnd	Alarm Type→ Video Loss	Event Type:Video Loss Event Action:Event End
Start - Video tampering of a video channel (e.g., "blind" camera / tilting camera)	Log_Event	BlindStart	Alarm Type→ Video Tampering	Event Type:Video Tampering Event Action:Event Start
End - Video tampering of a video channel (e.g., "un-blind" camera)	Log_Event	BlindEnd	Alarm Type→ Video Tampering	Event Type:Video Tampering Event Action:Event End
Modifying configurations (e.g., recording schedule).	Log_Event	Log_SaveConfig	System→Save Config	-
Triggering a tripwire detection rule (e.g., human detected).	Log_Event	InlAlmStart	Alarm Type→ Intelligent	Event Type:Tripwire Event Action:Event Start
AI detected a person.	Log_Event	InlAlmStart	Alarm Type→ Intelligent	Event Type:Face Detection Event Action:Event Start
AI recognized a person.	Log_Event	InlAlmStart	Alarm Type→ Intelligent	Event Type:Face Recognition Event Action:Event Start
Enrolled IP Camera Disconnected	Log_Event	NetAbortStart	Alarm Type→ Network Disconnection Event	Event Type:CAM Offline Alarm Event Action:Event Start
Enrolled IP Camera Detected	Log_Event	NetAbortEnd	Alarm Type→ Network Disconnection Event	Event Type:CAM Offline Alarm Event Action:Event End
Set recording type to "Auto" → Recording according to schedule	Log_Recctrl	Log_AutoRec	Record Mode→ Auto	Event Type: Record Mode
Set recording type to "Manual" → Continuous recording for 24H overriding schedule	Log_Recctrl	Log_ManualRec	Record Mode→ Manual	Event Type: Record Mode
Set recording type to "Close" → Disable recording.	Log_Recctrl	Log_ManualStop	Record Mode→ Close	Event Type: Record Mode
Reboot CCTV system.	-	-	System→Reboot	-
Shutdown CCTV system.	-	-	System→Shutdown	-
Sync System Time.	-	-	System→Sync System Time	-
View Live Footage.	-	-	-	-
Search Playback Video Recording.	-	-	Playback→Search Record	-
Play Playback Video Recording.	-	-	-	-
Attach USB stick.	-	-	Playback→Backup Device Found	-
Export Logs ("Backup").	-	-	Playback→Log Backup	-
Export Configurations ("Backup").	-	-	Playback→Config Backup	-
Monitor the health of the hard drives and detect any potential issues.	-	-	Storage→S.M.A.R.T	-
Identifying inserted hard disk.	-	-	Storage→Disk	-
Remove disk from system.	-	-	Storage→No Disk	-
Format CCTV system/ Format CCTV system with option "*Clear HDD db*" enabled	-	-	Storage→Format	-
Delete ("Clear") logs.	-	-	Clear Log→ Clear Log	-

APPENDIX B – Identified log types in the HIKVISION log records' research work

No.	Major Type		Minor Type		Parsing status
	Hex	Text	Hex	Text	Text
1	0x0300	Operation	0x4100	Power On	Parsed
2	0x0300	Operation	0x4200	Local: Shutdown	Parsed
3	0x0300	Operation	0x4300	Local: Abnormal Shutdown	Parsed
4	0x0300	Operation	0x5000	Local: Login	Parsed
5	0x0300	Operation	0x5100	Local: Logout	Parsed
6	0x0300	Operation	0x5200	Local: Configure Parameters	Log 'Details' field is currently not parsed
7	0x0300	Operation	0x5c00	Local: Initialize HDD	Partially Parsed
8	0x0300	Operation	0x6e00	HDD Detect	Partially Parsed
9	0x0300	Operation	0x7000	Remote: Login	Parsed
10	0x0300	Operation	0x7100	Remote: Logout	Parsed
11	0x0300	Operation	0x7600	Remote: Get Parameters	Parsed
12	0x0300	Operation	0x7700	Remote: Configure Parameters	Log 'Details' field is currently not parsed
13	0x0300	Operation	0x7800	Remote: Get Working Status	Parsed
14	0x0300	Operation	0x7900	Remote: Alarm Arming	Parsed
15	0x0300	Operation	0x7a00	Remote: Alarm Disarming	Parsed
16	0x0300	Operation	0x8000	Remote: Playback by Time	Log 'Details' field is currently not parsed
17	0x0300	Operation	0x8200	Remote: Initialize HDD	Partially Parsed
18	0x0300	Operation	0x8600	Remote: Export Config File	Parsed
19	0x0400	Information	0xa000	Time Sync.	Log 'Details' field is currently not parsed
20	0x0400	Information	0xa100	HDD Information	Parsed
21	0x0400	Information	0xa200	S.M.A.R.T. Information	Partially Parsed
22	0x0400	Information	0xa300	Start Record	Partially Parsed
23	0x0400	Information	0xa400	Stop Record	Partially Parsed
24	0x0400	Information	0xaa00	System Running State	Log 'Details' field is currently not parsed
25	0x0100	Alarm	0x0300	Start Motion Detection	Parsed
26	0x0100	Alarm	0x0400	Stop Motion Detection	Parsed
27	0x0100	Alarm	0x0500	Start Video Tampering	Parsed
28	0x0100	Alarm	0x0600	Stop Video Tampering	Parsed
29	0x0200	Exception	0x2200	Illegal Login	Parsed
30	0x0200	Exception	0x2400	HDD Error	Log 'Details' field is currently not parsed
31	0x0200	Exception	0x2700	Network Disconnected	Log 'Details' field is currently not parsed
32	0x0200	Exception	0x5400	Hik-Connect Offline Exception	Log 'Details' field is currently not parsed

APPENDIX C – Interpretation of the artifacts identified in the HIKVISION mobile app research work

OS|Artifact	Information Location	Information Format and Interpretation
Android|/databases/ezvizlog.db and iOS|/Documents/DCLOG/ YSDCLogItem.sqlite	Android| "*systemName*" column within the "***event***" table and iOS| "*systemName*" column of "YSDCLogItem" table	This column's values are of type *VARCHAR/TEXT*. The following values have been deciphered: - ***app_system_event***: the entry where this value is found should contain information about the mobile device (OS, etc.), the connection type, application start and stop time, etc. - ***app_video_p2p_pre/app_video_direct_pre***: the entries where these values are found should contain information about the CCTV system's WAN IP, serial number, local IP, etc. - ***GROUP***: the entry where this value is found should contain information about the user's actions ("*Live View*" or "*Playback*") when "*Hik-Connect*" platform is utilized to access the CCTV system (option "*Hik-Connect Domain*", "*Hik-Connect*" account, etc.). -***app_local_play***: the entry where this value is found should contain information about the user's actions ("*Live View*" or "*Playback*") when the CCTV system is directly accessed (option "*IP/Domain*") without utilizing "*Hik-Connect*" platform.
Android|/files/devmgr.user-ID{5}.sec.realm and iOS|/Documents/EZ_REALM/user-ID.realm	Android|"*DeviceConnectInfo*" table and iOS|"YSDeviceConnectionInfo" table	This table stores information about CCTV system's serial number, LAN IP and WAN IP.
	Android|"*DeviceHiddnsInfo*" table and iOS|"*YSDeviceHiddnsInfo*" table	This table stores information about the status of UPnP web port and server port.
	Android|"*DeviceInfo*" table and iOS|"*YSDeviceInfo*" table	This table stores information about CCTV system's model, firmware version and its current status (online/offline) along with the "*Hik-Connect*" user's account creation timestamp.
	Android|"*DeviceStatusInfo*" table and iOS|"*YSDeviceStatusInfo*" table	This table stores information about CCTV system's attached hard drives.
	Android|"*DeviceWifiInfo*" table and iOS|"*YSDeviceWifiInfo*" table	This table stores information about CCTV system's LAN IP, connection type and getaway.
	Android|"*Shareinfo*" table and iOS|"*YSDeviceShareInfo*" table	This table stores information about whether CCTV system is used by its bind "*Hik-Connect*" account (the original account who bound this CCTV system with the account) or a shared account (an account whose access to the CCTV system was granted by its bind account). If the value of "***isShared***" is "**1**" then this account is the bind one else if it is "**2**" this is a shared account.
Android|/shared_prefs/user-ID.xml		This file exists when "*Hik-Connect*" or "*Visitor Mode*" accounts are used. It stores XML variables/values. The following XML values have been deciphered: -"**USER_FIRST_LOGIN_TIME**": The value of this variable denotes the date when the user logged in the app (in UNIX epoch format). -"**PLAY_VIEW_SHORTCUT_BUTTON_COUNT**"/"**PLAY_VIEW_BUTTON** **_SEQUENCE _MAP**": These two variables were created when the user accessed CCTV system's Live View footage ("*Live View*" action). -"**PLAY_BACK_VIEW_SHORTCUT_BUTTON_COUNT**"/"**PLAY_BACK_**

		VIEW_BUTTON_SEQUENCE_MAP": These two variables were created when the user accessed CCTV system's stored recordings ("*Playback*" action).
Android|/shared_prefs/default.xml		This file stores XML variables/values. The following XML values have been deciphered: -"**LOGIN_MODE"**: The value of this variable indicates whether an account is currently logged in the app and which type of account is this. A value of "**0**" means no account is currently used, a value of "**1**" denotes the usage of a "*Hik-Connect*" account and a value of "**3**" signifies a "*Visitor Mode*" account. -"**PLAY_VIEW_SHORTCUT_BUTTON_COUNT**"/"**PLAY_VIEW_BUTTON** **_SEQUENCE _MAP"**: These two variables were created when the user accessed CCTV system's Live View footage ("*Live View*" action) without using any type of account. -"**PLAY_BACK_VIEW_SHORTCUT_BUTTON_COUNT**"/"**PLAY_BACK_** **VIEW_BUTTON_SEQUENCE_MAP"**: These two variables were created when the user accessed CCTV system's stored recordings ("*Playback*" action) without using any type of account.