Forensic Science International: Digital Investigation 45 (2023) 301560

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 USA - Proceedings of the Twenty Third Annual DFRWS Conference

IoT forensics: Analysis of a HIKVISION's mobile app

Evangelos Dragonas [a, *], Costas Lambrinoudakis [a], Michael Kotsis [b]

[a] Department of Digital Systems, University of Piraeus, 80 M. Karaoli & A. Dimitriou St., 18534, Piraeus, Greece
[b] Department of Informatics and Computer Engineering, University of West Attica, 28 Ag. Spyridonos St., Athens, 12243, Greece

ARTICLE INFO

Article history:

Keywords:
IoT forensics
Mobile forensics
HIKVISION
CCTV
com.connect.enduser
com.hikvision.hikconnect
Android
iOS
ALEAPP
iLEAPP

ABSTRACT

CCTV surveillance systems are ubiquitous IoT products. These CCTV systems can be remotely operated using either a mobile or a desktop application. HIKVISION is a well-known manufacturer of such devices that offers a variety of applications that allow remote usage of their products. Research regarding digital forensics of HIKVISION's CCTV systems is scarce and currently only limited to recovering video footage from the devices themselves, skipping all valuable artifacts that could reside within the applications' data that were utilized to access them. This unexplored piece of evidence is currently not parsed by either commercial or open source software yet it can hide vital information for a number of investigative questions. In this paper, a HIKVISION's mobile application is thoroughly analyzed, in both Android and iOS operating systems, in pursuit of evidentiary data that could reside within. Exploiting the findings of this study authors contributed to FOSS with the aim of assisting investigators with their examinations. In particular, they used their findings to develop relevant parsers for ALEAPP and iLEAPP.
© 2023 The Author(s). Published by Elsevier Ltd on behalf of DFRWS This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Statistics about IoT-connected devices worldwide indicate their number will potentially reach 17,08 billion by the end of 2025 and could go up as high as 29,42 by 2030 (Statista, 2019). Closed-Circuit Television (CCTV) surveillance systems are regarded as such devices and occupy a big part of the IoT world. CCTV systems can be found everywhere and as such regularly record important information related to crime activity. As a result, the digital forensic examination of these IoT devices is often critical to investigations. On the other hand, perpetrators may seek ways to tamper with this source of evidence in an attempt to hide their tracks. To complicate things even more for investigators, these CCTV devices can be remotely accessed and configured using either a mobile or a desktop application.

HIKVISION is a Chinese manufacturer of security cameras and other surveillance equipment. The company also offers a variety of applications, available for multiple operating systems, which can allow remote usage of their products. A recent analysis from Research and Markets (Research and Markets) considers HIKVISION among the global surveillance camera market leaders.

Despite its global market share, research regarding digital forensics of HIKVISION's CCTV systems is scarce and currently only limited to recovering video footage from the devices themselves, skipping all valuable artifacts that could reside within the applications' data that were utilized to access them. 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.

1.1. Research objectives

This study aims to deal with some of the aforementioned challenges. It focuses on proving that an end user has interacted with a HIKVISION's CCTV system by investigating its companion mobile application. Another goal is to verify the actions the user took while accessing the CCTV system. Contributing the results of this study to free and open-source software (FOSS) is its final objective.

1.2. How this paper contributes to digital forensics

To the best of the authors' knowledge, this is the primary account of someone investigating a mobile application that can be used to remotely access CCTV systems. The key takeaways of this study are summarized below.

* Corresponding author.
E-mail address: dragvag@ssl-unipi.gr (E. Dragonas).

https://doi.org/10.1016/j.fsidi.2023.301560
2666-2817/© 2023 The Author(s). Published by Elsevier Ltd on behalf of DFRWS This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

- The exploration of capabilities that HIKVISION's mobile application offer to end users.
- The presentation of artifacts that can be obtained from its forensic analysis on both Android and iOS operating systems (OS).
- Exploiting RAM to decrypt realm databases.
- Contribution to FOSS by integrating the results of this work into relevant parsers for ALEAPP (Brignoni, 2023a) and iLEAPP (Brignoni, 2023b).

1.3. How this paper is organized

The rest of this paper is organized as follows: Section 2 provides an overview of related work regarding digital forensics of both CCTV systems and companion applications of other IoT devices. In Section 3 the equipment used in this study as well as the methodology followed for the creation, acquisition, and analysis of the mobile app's data and RAM are displayed. Findings are presented in Section 4 and a discussion related to the results and limitations of this work takes place in Section 5. Finally, in Section 6 the paper is summarized and future research topics are shared.

2. Related work

2.1. CCTV forensics

Current research related to digital forensics of CCTV systems heavily focuses 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.

Han et al. (2015) comprehensively analyzed and interpreted the HIKVISION file system. They identified its internal structure and highlighted key areas within the file system. They also demonstrated a process that allowed access to video files stored within. Sandeepa et al. (2018) also reverse-engineered the file system from a HIKVISION digital video recorder (DVR). They proposed an algorithm for extracting video data from its file system.

Gomm et al. (2020) reviewed approaches to CCTV forensics and proposed a new forensic workflow for acquiring and analyzing artifacts from CCTV systems which they later applied in three case studies. One of the case studies involved the examination of a GANZ DVR which used AVTECH proprietary file system, a file system that was previously examined by some of the authors (Gomm et al., 2016). Tobin et al. (2014) explained a reverse engineering approach to recover and interpret data from the AVTECH CCTV file system using an 'eavesdrop' method.

2.2. IoT companion applications' forensics

Apart from the forensic analysis of CCTV systems, authors identified several studies which deal with mobile apps of other IoT devices. In this section, some of these studies are mentioned.

Dorai et al. (2018) examined Nest devices along with a Google Home Mini and tracked-down artifacts from their usage within an iOS logical backup. Authors also introduced Forensic Evidence Acquisition and Analysis System (FEAAS), a forensic tool which was used to report their findings.

Kim et al. (2020) researched data collected from companion apps, Web Interfaces, and APIs of Samsung SmartThings, Google Nest Hub and Kasa IoT appliances. In their work, they consolidated artifacts from all available evidence sources so as to determine their forensic value.

Epifani (2020) presented artifacts retrieved from the investigation of the iOS mobile application that is used with Apple HomePod and Apple HomeKit. Dragonas (2021) demonstrated artifacts from the analysis of the Android companion application of the Xiaomi IoT Ecosystem.

2.3. CCTV mobile applications' forensics

On the other hand, the authors could not find any published research related to the examination of companion applications that provide the ability to remotely operate CCTV systems.

In the following section, the equipment used and the methodology employed is introduced.

3. Equipment and methodology

3.1. Equipment

For this study, the authors employed several fresh mobile devices and two new HIKVISION Generation 4th X Hybrid Video Recorders (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.

Since the preliminary results of this work indicated there were no major differences between their artifacts, the authors 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 (Magisk Manager) whereas the iPhone was jailbreaked using palera1n (palera1n).

HIKVISION offers various mobile applications for both Android and iOS platforms (Hikvision App Store). 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 study was the first one. The criteria were that "*Hik-Connect - for End user*" was more regularly and recently updated. Nonetheless, the authors did also take a look at "*HiLookVision*" structure. The way both applications store their data is very similar. However, a complete analysis of "*HiLookVision*" application remains a topic of potential future work.

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 amount of conducted experiments. Ways of extracting a full file system (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 (Magnet ACQUIRE) for the Android and libimobiledevice (Libimobiledevice) 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 was utilized (X-Ways Forensics). Additionally, as the application partially stored data in SQLite and realm database formats, DB Browser for SQLite (DB Browser for SQLite) and Realm Studio

(Realm Studio) were used for viewing this information.

Frida (Frida), fridump3 (Rascagneres), and CyberChef (CyberChef) 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 1 and 2 respectively. HIKVISION's mobile application along with its versions examined in this study are listed in Table 3.

3.2. Methodology

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

3.2.1. Reconnaissance

During this phase, the authors familiarized themselves with some of "*Hik-Connect - for End user*" mobile application's capabilities. 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's app repository (Hikvision App Store). According to Play Store's statistics (Hik-Connect), the Android application has surpassed 5 million downloads worldwide. Both Android and iOS applications offered similar capabilities so the authors will elaborate on one of them (Android).

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.

Table 1
Hardware equipment used.

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)

Hik-Connect is a HIKVISION platform designed specifically to help customers operate its security products (Hik-Connect). 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 suboptions, the user must also supply the credentials of the desired CCTV system's user.
- "***Pyronix***": Pyronix is another manufacturer of security systems and technologies (Pyronix). Authors could not examine further this type as access to Pyronix services was required.
- "***Router***": This type is used when the CCTV system belongs to specific models of HIKVISION Network Video Recorders (NVRs). Authors could not examine further this type as they did not have access to such a device.

Table 2
Software equipment used.

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 3
Versions of HIKVISION's mobile application researched in this study.

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

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 Fig. 1. After getting a grasp of the mobile application's capabilities the authors began the Preparation/Collection phase.

3.2.2. Preparation/collection

During 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.

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). 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 authors performed multiple actions 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.

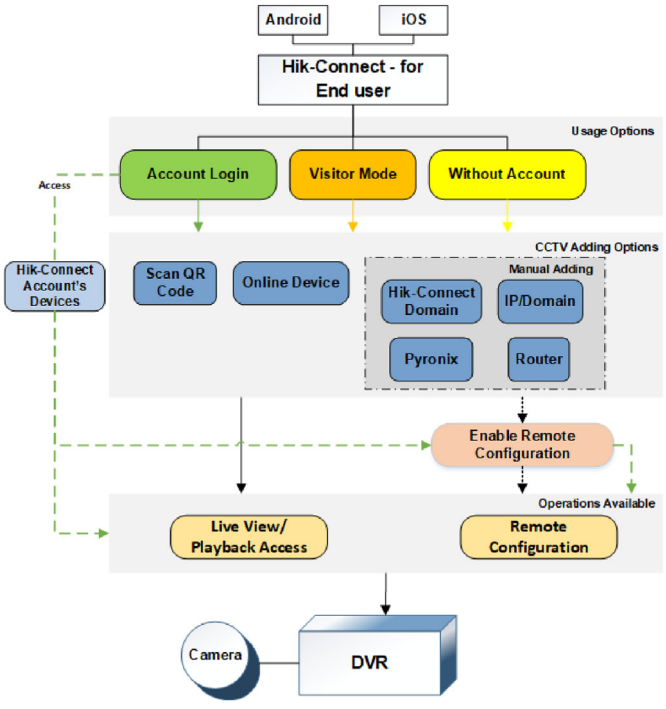

Fig. 1. Ways of accessing and configuring a CCTV system through HIKVISION's mobile application.

The diverse app features forced the authors to follow 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 Appendix A. 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.

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.

4. Results

4.1. Artifacts

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

4.1.1. Android app's data artifacts

Table 4 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

Table 4
Identified artifacts on Android OS.

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.)

"*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}*[1]*.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 to 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 4 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 4 can be found in Appendix B.

4.1.2. 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 study, this option was chosen.

After scrutinizing the output file, the decryption key was spotted as can be seen in Fig. 2. 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.

4.1.3. iOS app's data artifacts

Table 5 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

[1] user-ID{5}: denotes the first 5 characters of the user-ID.

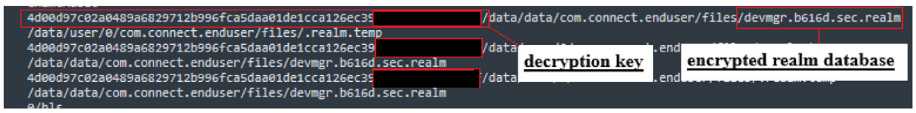

Fig. 2. Fridump3 strings command's output file includes the decryption key to the encrypted realm database.

Table 5
Identified artifacts on iOS OS.

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/Traffic Statistics.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”.

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 5 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 5 can be found in Appendix B.

4.1.4. 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.

4.2. App's behavior

In this section, authors describe how the application handles some of the aforementioned artifacts. 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”.

4.2.1. 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.

4.2.2. 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.

4.3. 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 Fig. 3). The keys within the red boxes in Fig. 3 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.

{"cn":1,"serial":"J10[redacted]","display_t":"11:45:41:778","err":0,
"1920*1088","rc":0,"screen":1,
"start_t":"11:45:41:594","stop_t":"11:45:54:852","via":1,"client
"appVer":"5.0.0.1125","osVer":"9","systemName":"app_local_play",
"11a527acf33f441b9[redacted]","g_uuid":
"f8809a5a-2ea1-[redacted]","lt":1670147154854,"lid":
"ad14daba-6e15-[redacted]","g_db":"main"}

Fig. 3. A user's "Live View" action created this entry within the "event" table of "ezvizlog.db" database.

- "**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.

4.4. Contributing to FOSS

Exploiting the findings of this study authors contributed to ALEAPP and iLEAPP software. In particular, authors developed SQLite queries for recovering evidentiary data from "ezvizlog.db", "image.db", "database.hik", and "YSDCLogItem.sqlite" databases. These queries were integrated into Python parsers that were used to extend the capabilities of these tools. Both queries and parsers are available in ALEAPP and iLEAPP official repositories.

An ALEAPP report of our parser's results is presented below (See Fig. 4). This concludes the Results section of this study.

5. Discussion

To the best of the authors' knowledge, no previous studies with reference to the analysis of a mobile application that can be used to remotely access CCTV systems were found yet the information that gets stored within them can be proven critical in certain investigations.

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

Among other things, our findings can help determine the user of the application, the IP of the CCTV system that was remotely

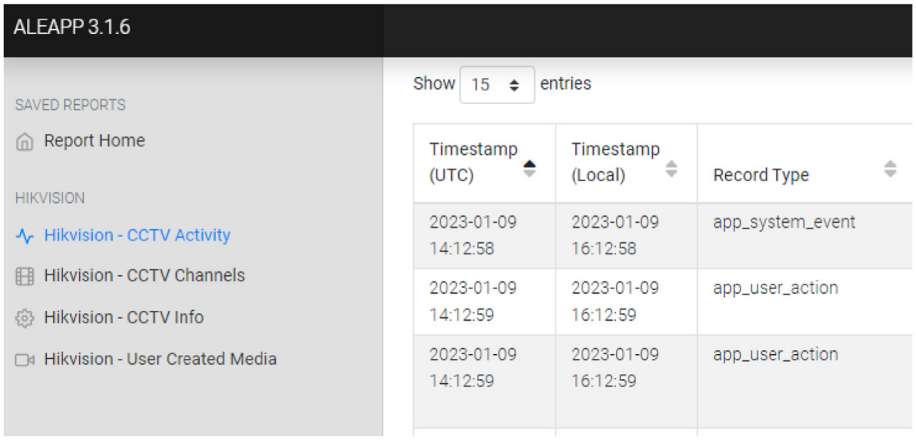

Fig. 4. An ALEAPP report of our parser's results.

accessed, and certain user actions (Live View/Playback/Create Media Files).

Additionally, our 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.

Using this study as a point of reference along with ALEAPP and iLEAPP reports, an examiner should be able to answer a number of questions that could remain unanswered up till now.

5.1. Admissibility

In general, the legal admissibility of CCTV evidence depends on the jurisdiction and the specific circumstances of the case. CCTV evidence can be admissible in court if it meets certain criteria such as its footage must be authentic, reliable, and accurate.

During our study, the user of HIKVISION's mobile application could not upload any kind of media files (images, video) to the CCTV system. She could only create and store media files originating from the CCTV recorded footage. This suggests that a user cannot fabricate CCTV footage directly from HIKVISION's mobile application. She could however fabricate stored media files on the mobile device using other applications. Furthermore, the user could not remotely wipe CCTV footage directly from HIKVISION's mobile application.

In a real investigation, the court should authorize the forensic analysis of both HIKVISION's mobile application and the CCTV system. This approach should help draw a safer conclusion.

5.2. Limitations

Nonetheless, this study has its own limitations. It does not take into account any of the evidentiary data that resides within the CCTV system itself. As suggested before, in a real investigation this source of evidence would not and should not be skipped. Correlating artifacts obtained from the analysis of both the mobile application and the CCTV system's log records could help draw more solid conclusions and attribute users' actions accordingly. Even though the authors did some preliminary examination of the XVR's log records their complete analysis is a subject of future work.

What is more, the authors did not examine the unallocated space of the employed mobile devices since they did not acquire their physical image. Unallocated space is another storage area where evidentiary data could reside and should not be omitted during an investigation.

6. Conclusions and future work

CCTV surveillance systems are considered ubiquitous IoT solutions which more often than not become crimes witnesses. These devices can be remotely accessed and configured using a plethora of applications. These applications however have not been forensically researched before leaving certain investigative questions unanswered. In this study, authors tried to fill a small part of this gap by analyzing HIKVISION's widely used mobile application. To further assist investigators authors contributed their findings to FOSS.

As part of their future work authors is interested in several topics. Firstly, they are interested in the examination of the other HIKVISION mobile application namely “HiLookVision”. Secondly, the authors are interested in the examination of HIKVISION's desktop applications designed for end users as well as more feature-rich HIKVISION's CCTV surveillance systems. Successfully interpreting artifacts from their forensic analysis could be proven useful to investigators. Lastly, the authors are interested in correlating artifacts retrieved from both applications' data and CCTV system's log records while tackling an “anti-forensics” scenario.

Appendix A

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

Appendix B

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.

References

CyberChef [Online]. Available: https://gchq.github.io/CyberChef/. Accessed 5 January 2023.

DB browser for SQLite [Online]. Available: https://sqlitebrowser.org/. Accessed 4 January 2023.

"Download Magisk Manager Latest Version 25.2 For Android 2022," . https://magiskmanager.com/ (accessed Jan. 04, 2023).

Frida - a world-class dynamic instrumentation toolkit [Online]. Available: https://frida.re/. Accessed 4 January 2023.

Hik-connect - for end user - apps on Google play. Play Store, [Online]. Available: https://play.google.com/store/apps/details?id=com.connect.enduser. Accessed 28 March 2023.

Hikvision app store [Online]. Available: https://appstore.hikvision.com/. Accessed 4 January 2023.

DynDNS [Online]. Available: https://account.dyn.com/. Accessed 5 January 2023.

"Magnet ACQUIRE" Magnet forensics [Online]. Available: https://www.magnetforensics.com/resources/magnet-acquire/. Accessed 5 January 2023.

"Libimobiledevice · A cross-platform FOSS library written in C to communicate with iOS devices natively" libimobiledevice [Online]. Available: https://libimobiledevice.org/. Accessed 5 January 2023.

palera1n [Online]. Available: https://palera.in. Accessed 4 January 2023.

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

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

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

Dorai, G., Houshmand, S., Baggili, I., 2018. I know what you did last summer: your smart Home internet of things and your iPhone forensically ratting you out. In:

Proceedings of the 13th International Conference on Availability. Reliability and Security, New York, NY, USA, pp. 1–10. https://doi.org/10.1145/3230833.3232814.
Forensic analysis of Apple HomePod & Apple HomeKit environment w/mattia epifani - SANS DFIR summit [Online Video]. Available: https://www.youtube.com/watch?v=D8AOXCBkaTY. Accessed March 28, 2023.
Forensic analysis of Xiaomi IoT Ecosystem w/evangelos dragonas - SANS DFIR summit [Online Video]. Available: https://www.youtube.com/watch?v=4oVfHinPlz0. Accessed March 28, 2023.
Gomm, R., Le-Khac, N.-A., Scanlon, M., Kechadi, T., 2016. Analytical Approach to the Recovery of Data from CCTV File Systems.
Gomm, R., Brooks, R., Choo, K.-K.R., Le-Khac\kappa, N.-A., Hew, K.W., 2020. "CCTV forensics in the big data era: challenges and approaches" at. Cyber and Digital Forensic Investigations : A Law Enforcement Practitioner's Perspective 74, 109–140. Springer International Publishing.
Han, J., Jeong, D., Lee, S., 2015. "Analysis of the HIKVISION DVR File System" at Digital Forensics and Cyber Crime, vol. 157. Springer International Publishing, pp. 189–199.
Hik-Connect. HIKVISION, [Online]. Available: https://www.hik-connect.com/. Accessed 4 January 2023.
Kim, S., Park, M., Lee, S., Kim, J., 2020. Smart Home forensics—data analysis of IoT devices. Electronics 9 (8), 1215. https://doi.org/10.3390/electronics9081215.
Rascagneres, P. fridump3 [Online]. Available: https://github.com/rootbsd/fridump3. Accessed 4 January 2023.
Research and Markets ltd., Sep. 2022. 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. https://www.researchandmarkets.com/reports/5239813/global-surveillance-camera-market-analysis-by. (Accessed 19 December 2022).
Sandeepa, S., Reyaz, A., Silpa, M., 2018. An Efficient Approach to Recover CCTV Video from Proprietary DVR File System" at 2018 International CET Conference on Control, Communication, and Computing. IC4).
Pyronix [Online]. Available: https://www.pyronix.com/. Accessed 5 January 2023.
Statista. IoT connected devices worldwide 2019-2030 [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. Accessed 19 December 2022.
Tobin, L., Shosha, A., Gladyshev, P., 2014. Reverse engineering a CCTV system, a case study. Digit. Invest. 11 (3), 179–186.
"X-Ways Forensics: Integrated Computer Forensics Software." https://www.x-ways.net/forensics/index-m.html (accessed Jan. 04, 2023).