Journal of Cybersecurity and Privacy

MDPI

Article

Investigating the Privacy and Security of the SimpliSafe Security System on Android and iOS

Shinelle Hutchinson [1,†], Miloš Stanković [1,†], Samuel Ho [1], Shiva Houshmand [2] and Umit Karabiyik [1,*]

[1] Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA
[2] Department of Mathematics and Computer Science, Santa Clara University, Santa Clara, CA 95053, USA
* Correspondence: umit@purdue.edu
† These authors contributed equally to this work.

Abstract: The emergence of the Internet of Things technologies and the increase and convenience of smart home devices have contributed to the growth of self-installed home security systems. While home security devices have become more accessible and can help users monitor and secure their homes, they can also become targets of cyberattacks and/or witnesses of criminal activities, hence sources of forensic evidence. To date, there is little existing literature on forensic analysis and the security and privacy of home security systems. In this paper, we seek to better understand and assess the forensic artifacts that can be extracted, the security and privacy concerns around the use of home security devices, and the challenges forensic investigators might encounter, by performing a comprehensive investigation of the SimpliSafe security system. We investigated the interaction of the security system with the SimpliSafe companion app on both Android and iOS devices. We analyzed the network traffic as the user interacts with the system to identify any security or privacy concerns. Our method can help investigators working on other home security systems, and our findings can further help developers to improve the confidentiality and privacy of user data in home security devices and their applications.

Keywords: Android; digital forensics; iOS; IoT; mobile forensics; network forensics; home security; privacy; security; smart home

Citation: Hutchinson, S.; Stanković, M.; Ho, S.; Houshmand, S.; Karabiyik, U. Investigating the Privacy and Security of the SimpliSafe Security System on Android and iOS. *J. Cybersecur. Priv.* **2023**, *3*, 145–165. https://doi.org/10.3390/jcp3020009

Academic Editors: Mario Antunes and Carlos Rabadão

Received: 20 February 2023
Revised: 3 April 2023
Accepted: 4 April 2023
Published: 7 April 2023

1. Introduction

Home security systems have been growing and are expected to grow to USD 84.4 billion by 2027 [1]. While home security and smart home devices provide convenience and ease of mind for users, their associated data are also increasingly used as forensic evidence in criminal cases [2,3]. Although home security devices help secure and monitor homes, they can become great sources of forensic evidence. With the rise of self-installed home security devices, the important question to ask is how well these devices protect user data. Home security systems are categorized into (1) traditional, professionally installed and monitored systems, (2) self-installed and professionally monitored, and (3) self-installed and self-monitored smart IoT devices. The do-it-yourself self-installed systems are predicted to have the fastest growth and surpass USD 32 million by 2030 [4]. The growth is due to an increase in awareness of home security systems, the emergence of Internet of things (IoT) technologies, and easy-to-use do-it-yourself (DIY) home security systems. Among the self-installed home security systems, SimpliSafe [5] is usually ranked in the top three favorite systems [6].

In this paper, we investigate the SimpliSafe home monitoring system due to its high popularity, focusing on the interaction between the companion app and devices, as well as a network analysis while the base device connects to servers. SimpliSafe is a self-installed smart home security system that includes monitoring sensors and cameras.

The main contributions of our study are as follows:

- We conducted a mobile forensic analysis of the SimpliSafe home security app on both Android and iOS smartphones.
- We performed a forensic analysis of the network traffic generated when a user interacts with the SimpliSafe home security system.
- We identified any privacy and security concerns that arose due to the way the SimpliSafe home security app stores data on smartphones and transmits data over a network.
- We provide a forensic road map to investigators tasked with examining this and similar home security systems.

This paper is organized as follows. In the next section, we share background information and related works. In Section 3, we explain the methodology followed in this paper. Section 4 describes our findings. In Section 5, we discuss our results and challenges. Finally, in Section 6, we conclude our analysis and discuss future work.

2. Related Work

In this section, we discuss some of the related work in mobile application forensics of smart home devices, smart home forensic investigations, and home security system investigations. An overview of the reviewed research and its shortcomings are presented in Table 1.

There have been several previous efforts in recovering forensic artifacts from smart home IoT devices. For example, Hutchinson et al. [7] conducted a forensic investigation of various smart home devices in a lab setup and discussed possible threat scenarios. They were able to recover several private pieces of information such as email, full name, OAUTH credentials, etc., from the smart home devices and their companion apps and discussed the privacy concerns and security vulnerabilities when using these smart home devices. Chung et al. [8] analyzed client-centric and cloud artifacts stored within companion apps of the Amazon Echo. Dorai et al. [9] investigated the forensic artifacts produced by Nest ecosystems such as Nest thermostats and Nest cameras. Other studies have looked at smart devices' ecosystems and their interaction with each other. In [10], Hutchinson et al. investigated the August Smart Doorbell Cam and Lock devices and the interaction between these devices to determine what information could be acquired.

Although forensic investigations into home security systems and specifically SimpliSafe has been limited, there are a few previous works that have investigated security and privacy issues that we describe in this section. Janes et al. [11] evaluated the susceptibility of 19 common security cameras and doorbells to a persistent attack after access revocation, in situations when multiple users share a single account. Specifically, they looked at design flaws that failed to revoke a user's access that was requested to be removed by the account owner. In their study, they also included SimpliSafe Doorbell and Camera, and their results showed that the video stream on the Android companion app was still available for over 30 min after the account's password was changed. OConnor et al. [12] studied the design and implementation flaws of twenty smart home device companion apps. Their results showed that sixteen of these vendors suffered from some form of a design flaw that failed to properly validate certificates or protect the integrity of the message traffic. In particular, they showed that they were able to manipulate or clear alarm log files for SimpliSafe.

Through IoT and network forensics, Hutchinson et al. [7] also investigated the network traffic for over 10 IoT smart devices including the Google Nest Hub Max, Amazon Fire TV Cube, and the August smart lock and smart doorbell [10]. By investigating both the mobile applications and network traffic of these devices, the authors highlighted the serious security risks that threatened these devices' users. For instance, although some of the information found such as zip code, the make and model of the devices used, etc., may not be useful to a forensic investigation, they can be used as reconnaissance to gain further access into the home network which can lead to spear-phishing or blackmail.

Additionally, ref. [13] showed how Wireshark could be utilized for network protocol diagnosis and aid in network forensics. This was done by conducting multiple attack scenarios such as covert FTP and IRC channels, ICMP-based attacks, and distributed denial

of service (DDoS) attacks while running Wireshark concurrently to capture network traffic. The traffic was then analyzed to identify the attack vectors.

In [14], the authors used recognizable network traffic patterns to predict incoming distributed denial-of-service (DDoS) attacks. More specifically, they used Wireshark's I/O graph tool to discover five unique network traffic patterns that emerged when their home IoT environment was under a DDoS attack. The authors also provided guidelines to filter for specific traffic such as the TCP error flags and protocols used in Wireshark.

Table 1. Summary of Literature Review and Shortcomings

Article	Objectives	Methods and Techniques	Shortcomings
Hutchinson et al. [7]	To determine what data from IoT devices can be recovered, how to recover the data, and where these data reside.	The authors created an IoT forensics laboratory. They used XRY to create a physical image and XAMN to analyze the image for artifacts and evidence of privacy leaks.	The authors investigated individual home security devices such as an August Smart Lock Pro and August Smart Doorbell Pro but not a smart home monitoring system.
Chung et al. [8]	To investigate methods for digital forensics pertaining to the IVA Alexa's ecosystem.	The authors proposed a new integrative approach combining cloud-native and client-centric forensics for the Amazon Alexa ecosystem. They also introduced an implementation, CIFT, to acquire native artifacts from Alexa and analyze local artifacts from companion clients.	The authors did not perform their investigation on the hardware level of the Alexa-enabled devices. They also did not perform memory forensics for volatile artifacts.
Dorai et al. [9]	To examine the logical backup structure of an iPhone used to control a Nest thermostat, Nest indoor camera, and a Nest outdoor camera.	The authors built an open-source forensic tool called Forensic Evidence Acquisition and Analysis System (FEAAS), that consolidated evidentiary data into a readable report that could infer user events.	The study was only limited to iPhones and focused on data that were logically acquired from the mobile device, which meant that it only worked if data had not been deleted from the phone under examination.
Hutchinson and Karabiyik [10]	To determine what type of data forensic investigators may be able to recover about the August Smart Doorbell Pro and the August Smart Lock Pro, with their controlling app, August Home.	The authors used Magnet AXIOM Examiner and MSAB XRY to examine artifacts acquired from imaging one iOS and two Android smartphones.	The authors investigated two individual IoT devices (August Smart Lock Pro and August Smart Doorbell Pro) but not a smart home monitoring system.
OConnor et al. [12]	To better understand IoT security and privacy by studying the design flaws of this distributed communication channel for smart home devices.	The authors implemented a smart home lab environment with devices from 20 different vendors to explore the severity and pervasiveness of attacks against IoT devices.	The authors showed that they were able to manipulate or clear alarm log files for SimpliSafe. However, they only focused on whether the attack was successful or transparent. They did not investigate any recoverable artifacts related to user interactions with the system.
Ndatinya et al. [13]	To demonstrate how Wireshark can be applied in network protocol diagnosis and can be used to discover traditional network attacks.	The authors used Wireshark to identify certain types of network attacks that resulted in unusual activities as well as present case studies for typical network attacks by using Wireshark.	The authors found that Wireshark was one of the best open-source packet analyzers available. However, Wireshark can only analyze packet captures and network traffic. It does not have intrusion detection and network manipulation capabilities.
Ho et al. [14]	To discover network traffic patterns that emerge when IoT devices are under a DDoS attack.	The authors used LOIC and Slow Loris to perform a DDoS attack on the IoT devices. They used Wireshark to capture and examine the network packet captures while the attack was running	The authors only used Wireshark to analyze the network packet captures. Different software and metrics could be used to conduct both the attacks and the investigation processes.

3. Methodology

There is little to no forensic work related to IoT home security systems such as SimpliSafe, which leaves a wide area of possibilities for research stretching from the privacy and security of these systems to their potential for compromise. The aim of our study was to determine how much evidence can be obtained during a forensic investigation involving such a security system, particularly concerning the ability to recover artifacts related to user interactions with the system, alert notifications, and camera images. This information can be helpful in situations where a crime has occurred and investigators are looking for digital evidence from the security system. We were also interested in comparing the type and number of forensic artifacts that could be recovered from both operating systems investigated. To accomplish these goals, the SimpliSafe security system was forensically investigated via its controlling application on Android and iOS smartphones as well as the system's network communications. Similar methodologies were used in [7,10]. The result of our study is also useful in identifying any privacy and security concerns that arise for the homeowner as a result of using such systems.

3.1. Device Setup and App Installation

A typical SimpliSafe security system includes a base station which is the heart of the system that all sensors connect to. The base station sounds the alarm or alerts the monitoring station when a sensor is triggered. It also contains a wireless keypad that allows the user to arm or disarm the system and change settings. The system comes with a companion app where the user can monitor camera images or change settings. The SimpliSafe security system allows users to subscribe to monitoring services which enable users to record and store camera feeds to the cloud or connect the system directly to emergency response services. In this study, we used the free (unmonitored) subscription plan in which the user can only monitor everything through the app, but the system was not connected to an emergency response service. In our setup, we used the base station, the wireless keypad, an indoor security camera, two entry sensors, and a motion sensor (see Figure 1). All SimpliSafe devices (hub, keypad, sensors, and camera) were new so there was no need for reverting them to factory settings. Prior to activating the system, a Google email address (p****lab@gmail.com) was used to set up the SimpliSafe account. We also obtained a SIM card for cellular connection and the plan that needed to be associated with the security system.

Figure 1. The SimpliSafe security system, from left to right: base station, keypad, motion sensor, two entry sensors, and camera.

3.2. Lab and Scenario Setup

The physical location of the security system was within a single research lab. We formulated a burglary scenario to guide our investigation and set up the devices in locations to represent a one-story home with a basement. The location of devices in this scenario can be seen in Figure 2. Our burglary scenario involved someone breaking into the front door (entry sensor #1), walking past the indoor security camera in the living room, opening the basement door (entry sensor #2), triggering the motion sensor in the basement, and breaking into a safe stored in the basement.

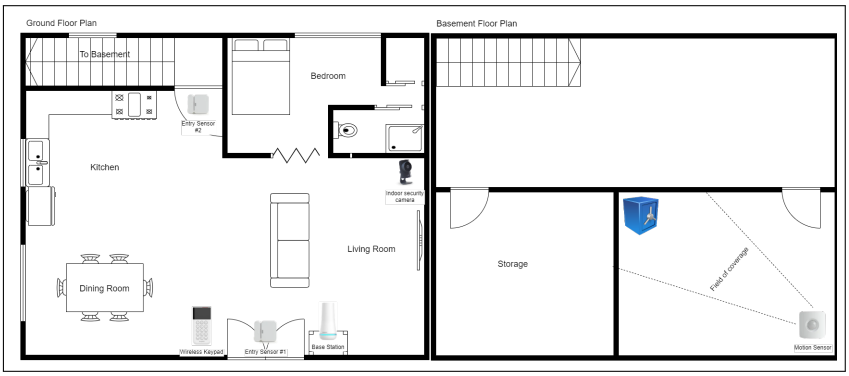


Figure 2. Location of security sensors for the burglary scenario.

3.3. Forensic Process

Figure 3 illustrates the research methodology followed in this study including the processes to populate, acquire, and analyze the relevant apps on both smartphones. Similarly, Figure 4 depicts the network and communication links among all devices used in the study. In this scenario, motion and entry sensors, as well as the keypad, were directly connected to the base station which was wirelessly connected to the access point provided by the laptop. The laptop acted as the bridge between the Internet and the local network, allowing us to capture the network traffic using Wireshark and further analyze the packets. The laptop with the Nmap [15] software allowed us to keep track of the IP addresses assigned to the base station, camera, and smartphones. Additionally, Figure 4 shows the indoor camera being connected directly to the laptop and not to the base station like other sensors. This is due to the camera's ability to operate independently of the base station and sensors.

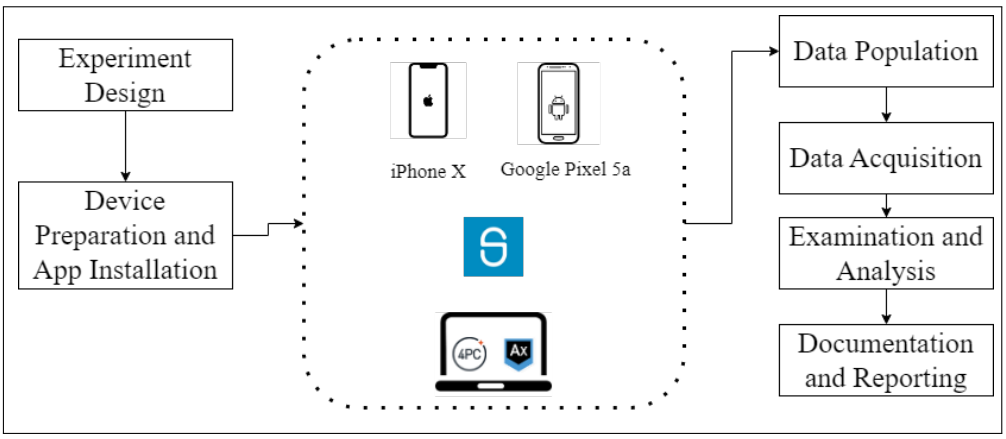


Figure 3. Research methodology used in the study.

3.3.1. Data Population

During the data population phase, we made sure to use test accounts that were never previously associated with the SimpliSafe system. The steps to populate our experimental devices with user data included (1) factory resetting the iPhone X (model: A1865) to default, (2) rooting and verifying root access on the Google Pixel 5a smartphone using a root checker app, and (3) entering data in devices according to the National Institute of Standards and Technology (NIST) [16] guidelines for populating a mobile device. This meant that we installed the SimpliSafe app from the App Store and the Google Play Store on both devices. A short summary of the steps we performed for the data population on both apps is as follows:

- We created an account and signed into the app.
- We set up the SimpliSafe devices on the account. This included connecting the base station and camera to the Internet.
- We interacted with the system by changing the alarm mode, **OFF**, **HOME**, or **AWAY**, via the app on the phone and the keypad.
- We triggered the alarm via each sensor.
- We viewed the camera feed from the app.

For the full timeline of the actions please refer to Appendix A.1 for Android and Appendix A.2 for iOS.

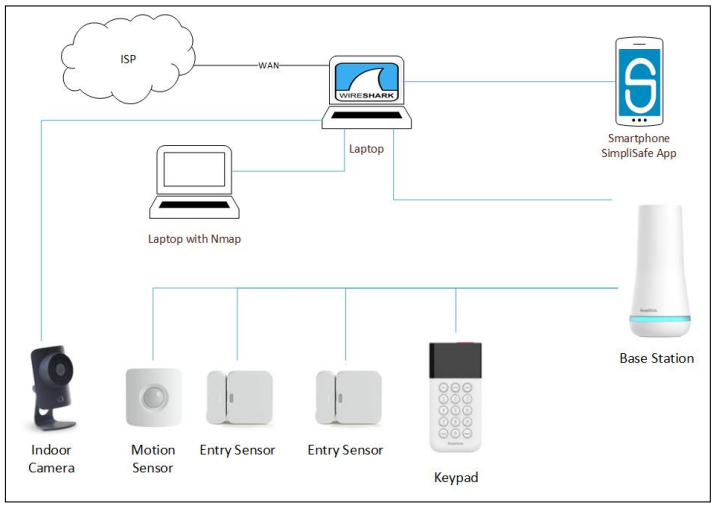


Figure 4. Network diagram showing devices and communication links.

3.3.2. Data Acquisition

Once the data population was completed, we immediately created the forensic image of the smartphone using either Cellebrite UFED 4PC [17] to acquire the iPhone X (advance logical image) and Magnet AXIOM Process [18] to acquire the Google Pixel 5a (logical image). Cellebrite UFED 4PC is equipped with the checkra1n exploit which was capable of jailbreaking our test iPhone. Similarly, because we rooted the Google Pixel 5a device, we expected full access to the file system. However, neither Cellebrite nor Magnet was able to provide a full/physical image of this Pixel device. All the software tools used throughout the process are shown in Table 2.

Table 2. List of Software Used and their Purpose.

Software Name	Software Version	Usage
Magnet AXIOM Acquire	2.57.0.32014	Acquire Evidence
Magnet AXIOM Process	6.9.0.34051	Acquire Evidence
Magnet AXIOM Examine	6.9.0.34051	Analyze Evidence
Cellebrite UFED 4PC	7.42.0.82	Acquire Evidence
Cellebrite Physical Analyzer	7.42.0.50	Acquire and analyze evidence
Cellebrite Reader	7.42.0.50	Analyze Evidence
Wireshark	4.0.3	Acquire and analyze evidence
Splunk Enterprise	9.0	Analyze Evidence
Checkra1n	0.12.4 (Beta)	Jailbreaking iPhone
Root Checker	6.5.0	Confirming root on Android
SimpliSafe App (Android)	4.61.0	Data population
SimpliSafe App (iOS)	2078.52.0	Data population

3.3.3. Examination and Analysis

We performed the examination and analysis of both forensic images using Magnet AXIOM Examine as this commercial tool suite has the capability to process both `.tar` (Google Pixel 5a) and `.zip` (iPhone X, A1865) image formats.

All artifacts discussed hereunder were recovered from the apps' packages. The iOS app package was found in the `\private\var\mobile\Containers\Data\Application\8E91EBEE-4046-4220-86EE-94CAA7DF32FD` and the Android app package was found in the `\data\data\com.simplisafe.mobile` folders in the respective file systems.

3.4. Network Traffic Capture

Wireshark, an open-source packet analyzer was used to collect network traffic during the data acquisition phase. Wireshark was selected specifically for this investigation as it provides a packet-by-packet view of network traffic [19]. The mode was initially set to OFF in the SimpliSafe application. The network capture was started in live view, then the mode was switched to Away, and the network capture was stopped. This scenario was repeated 10 times to ensure accuracy and consistency in the network traffic collected. Wireshark's built-in "I/O Graphs" tool was used to graphically represent the data for a visual information analysis. Each capture was converted into an I/O graph and then analyzed to see if there were any unexpected traffic patterns.

Additionally, Splunk [20] was used to further analyze the packet captures. Specifically, "PCAP Analyzer for Splunk" [21] was used as it uses the tshark component to convert pcap files into readable `csv` files. This allowed for the creation of many graphs including but not limited to bytes transferred by conversation, TCP flags over time, packet count by protocol, and TCP errors over time by source IP. The network traffic was then analyzed to see if there were any unexpected or outlying traffic patterns. If there was any unanticipated traffic detected, it could be a sign of a vulnerability or an insecure network.

The security of the network traffic was also analyzed to see if the levels of encryption and encoding were enough to prevent us from reading plaintext or unencrypted data. Specifically, we looked into the digital signature, key exchange, and session key generation protocols using Wireshark itself.

4. Results

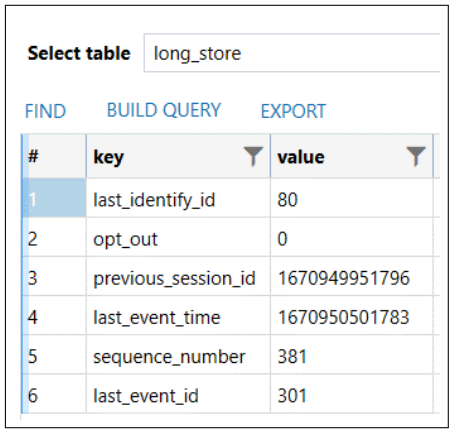
After completing our comprehensive digital forensic analysis of both Android and iOS images, we identified all forensically relevant artifacts that could be extracted and would benefit forensic investigators. In this section, we present our findings in detail.

4.1. Android Findings

Most notably, the recovered Android artifacts may help paint a picture of when and where the user used the app or the system, respectively. Session-related logs were recovered from both the `\app_bugfender` folder and the `\databases\com.amplitude.api` database. Figure 5 depicts the content of the long_store table within this database. These logs include session, device, and user ID numbers, network connection, failure logs, and session start and end timestamps, among others (Figure 6).

The `\shared_prefs\FirebaseHeartBeatW0RFRkFVTFRd+MTo2MDgzNjI2NDEwNzY6YW5kcm9pZDplZjYzOGI5ZGUxZWI2ZWU3.xml` file held the last date the app was used as well as a record of each day the app was used on the current device. Since the testing took place over two days, we were only able to confirm this list recorded at least two days of app usage (see Figure 7). Similarly, the first time the app was opened on the current device and the last time the app was paused (i.e., put into the background) could be recovered from the `\shared_prefs\com.google.android.gms.measurement.prefs.xml` file.

Two of the more relevant .xml files found within the `\shared_prefs` folder were the *SS_MISC.PREFERENCES.xml* file and the *SS_TOKENS.PREFERENCES.xml* file (see Figure 8). With these two files, investigators were able to identify (1) current_location_sid, (2) the number of times the user viewed the live camera feed, (3) the last time the app was opened, (4) the SID of the monitoring location, (4) the name of the camera that was set up, (5) the SID of the camera's location, and (6) the user's user ID and email address.



Select table long_store

FIND BUILD QUERY EXPORT

#	key	value
1	last_identify_id	80
2	opt_out	0
3	previous_session_id	1670949951796
4	last_event_time	1670950501783
5	sequence_number	381
6	last_event_id	301

Figure 5. Session-related log showing the timestamp of the last event that occurred on the app as well as the previous session ID.

{"event_type":"SocketLink_Connected","timestamp":
1670949275330,"user_id":"5300013","device_id":
"e59c1ca2-f7e8-446f-adc5-b0fb1d020b7cR","session_id":
1670949275329,"uuid":"43809544-2e11-4d7d-9f10-5e73a5f9d34e",
"sequence_number":354,"version_name":"4.61.0","os_name":
"android","os_version":"11","api_level":30,"device_brand":
"google","device_manufacturer":"Google","device_model":
"Pixel 5a","carrier":"Mint","country":"US","language":"en",
"platform":"Android","library":{"name":"amplitude-android",
"version":"2.38.3"},"api_properties":{"tracking_options":{
"ip_address":false,"lat_lng":false},"androidADID":
"30056f35-a734-4ea8-a0a8-e82699eed01d","limit_ad_tracking":
false,"gps_enabled":true},"event_properties":{},
"user_properties":{},"groups":{},"group_properties":{}}

Figure 6. Session-related log showing when a socket connection was made along with the user ID associated with that action.

```
FIND

<?xml version="1.0" encoding="utf-16" standalone="yes"?>
<map>
    <long name="fire-count" value="2" />
    <string name="last-used-date">2022-12-13</string>
    <long name="fire-global" value="1670869017686" />
    <set name="fire-core/20.2.0 fire-perf/20.2.0 android-min
        -sdk/24 fire-fiam/20.2.0 kotlin/1.7.20 fire-analytics
        /21.2.0 fire-fcm/23.1.0 fire-abt/21.1.0 fire-fiamd/20
        .2.0 device-model/barbet device-name/barbet android
        -installer/com.android.vending fire-installations/17.1
        .0 fire-android/30 device-brand/google fire-rc/21.2.0
        android-platform/ fire-transport/18.1.7 fire-cls/18.3
        .1 android-target-sdk/33">
        <string>2022-12-13</string>
        <string>2022-12-12</string>
    </set>
</map>
```

Figure 7. Record of the last day the app was used including the list of (at least) the last two days.

```
<?xml version="1.0" encoding="utf-16" standalone="yes"?>
<map>
    <long name="last_launched_time" value="1670947996830" />
    <string name="one_monitoring_locations">5962533</string>
    <string name="one_camera_location_name">U[redacted]LAB</string>
    <string name="userId">5300013</string>
    <string name="email">p[redacted]lab@gmail.com</string>
    <string name="one_camera_location">5962533</string>
</map>
```

Figure 8. A redacted record of the content of the *SS_TOKENS.PREFERENCES.xml* file.

Investigators may be able to use current_location_sid to elicit the physical address from SimpliSafe (e.g., the home address) where the system is being used. This may be necessary if the smartphone is the only piece of evidence that investigators have access to during a time-sensitive investigation, such as a kidnapping case.

Other usage-related artifacts were also recovered, including the timestamps of when pop-up notifications were shown to the user on the app (`\shared_prefs\SS_USER_ACTIONS.PREFERENCES.xml` file), authentication keys/tokens, and IV's for the SimpliSafe app along with expiration timestamps (`\shared_prefs\com.auth0.authentication.storage.xml` file).

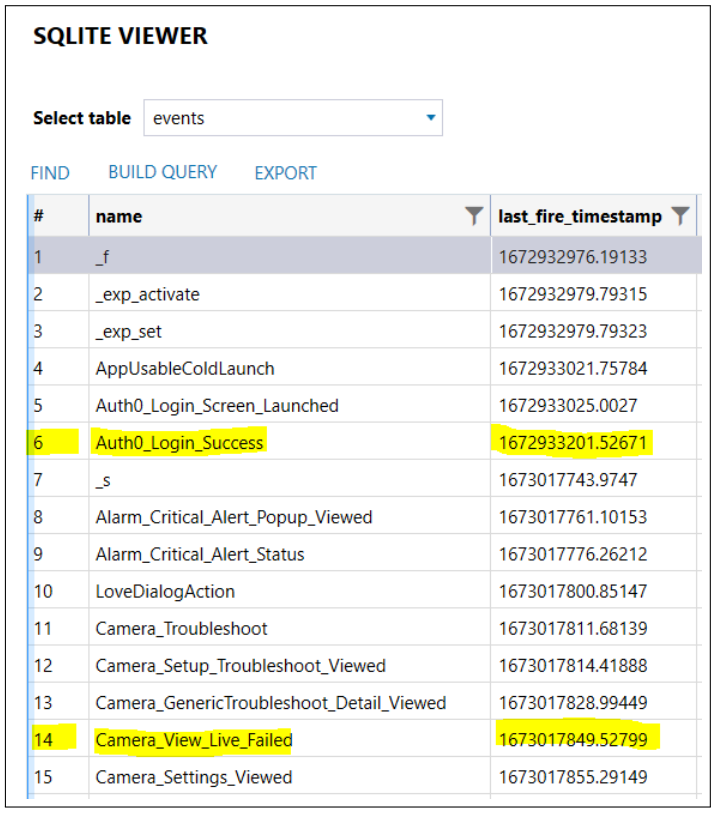
4.2. iOS Findings

A variety of methods and tools were utilized to retrieve the information in this section. For the data acquisition, Cellebrite UFED 4PC was used and the generated `.dar` image file was placed in Magnet AXIOM Process to create the case which was later used for the examination and analysis via Magnet AXIOM Examine. In order to make it easier to display the findings, the initial portion of the path `\private\var\mobile\Containers\Data\Application\8E91EBEE-4046-4220-86EE-94CAA7DF32FD` is removed in the following discussions.

The first finding for iOS was at the `\Library\ApplicationSupport\Google\Measurement` location within the `google-app-measurement.sql` database and events table. This table shows events throughout the data population (see Figure 9). The highlighted item's (Auth0_Login_Success) timestamp matched our timeline (Appendix A.2) when we logged into the app on 5 January 2023 around 10:37. Similarly, Camera_View_Live_Failed matched the timeline when the camera was accessed unsuccessfully. These are only two examples while the table contained 49 records in total.

Various sessions were found in the `\Library\Caches\com.bugfender.BugfenderSDK\BFPersistedStringQueues` folder. We found two sessions, one of less value and one containing valuable information for the investigation. Within the *session-24842698825* folder and *session.json* file, information such as udid, key, device type,

OS version, and starting time of the session was found (shown in Figure 10). This time matched the time when we signed into the application for the first time.



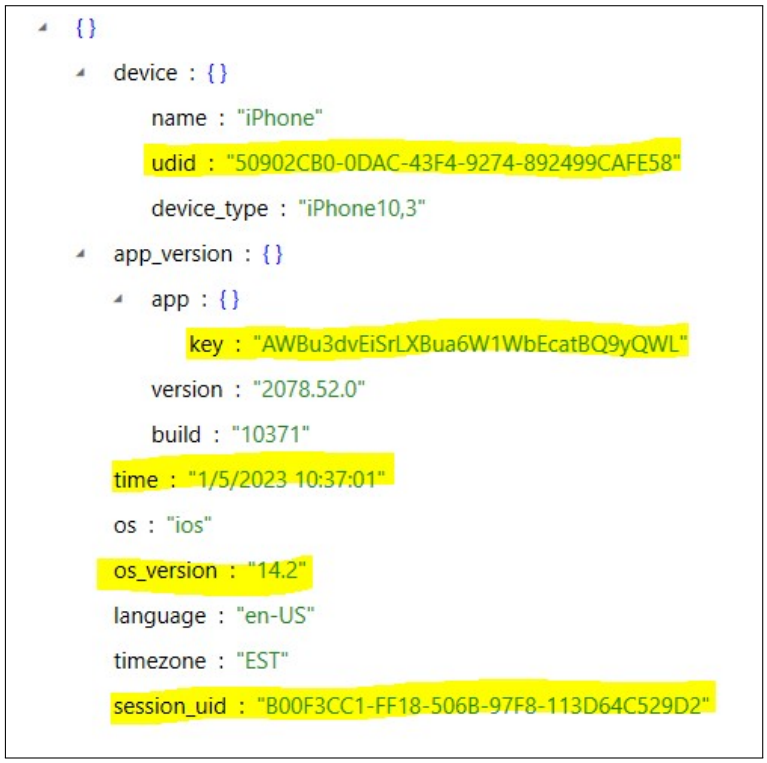
SQLITE VIEWER

Select table events

FIND BUILD QUERY EXPORT

#	name	last_fire_timestamp
1	_f	1672932976.19133
2	_exp_activate	1672932979.79315
3	_exp_set	1672932979.79323
4	AppUsableColdLaunch	1672933021.75784
5	Auth0_Login_Screen_Launched	1672933025.0027
6	Auth0_Login_Success	1672933201.52671
7	_s	1673017743.9747
8	Alarm_Critical_Alert_Popup_Viewed	1673017761.10153
9	Alarm_Critical_Alert_Status	1673017776.26212
10	LoveDialogAction	1673017800.85147
11	Camera_Troubleshoot	1673017811.68139
12	Camera_Setup_Troubleshoot_Viewed	1673017814.41888
13	Camera_GenericTroubleshoot_Detail_Viewed	1673017828.99449
14	Camera_View_Live_Failed	1673017849.52799
15	Camera_Settings_Viewed	1673017855.29149

Figure 9. Events recorded by the SimpliSafe app on the iOS device.



{}
device : {}
name : "iPhone"
udid : "50902CB0-0DAC-43F4-9274-892499CAFE58"
device_type : "iPhone10,3"
app_version : {}
app : {}
key : "AWBu3dvEiSrLXBua6W1WbEcatBQ9yQWL"
version : "2078.52.0"
build : "10371"
time : "1/5/2023 10:37:01"
os : "ios"
os_version : "14.2"
language : "en-US"
timezone : "EST"
session_uid : "B00F3CC1-FF18-506B-97F8-113D64C529D2"

Figure 10. Session details recovered from the iOS SimpliSafe app.

Moreover, the `\Library\Caches\com.bugfender.BugfenderSDK\session-24842698825\logs` folder contained logs stored throughout the session. The log records started right after the first login. The logs were consistent with the app being opened and used. For example, on 5 January 2023, logs were only created during the times of the data population (see Figure 11).

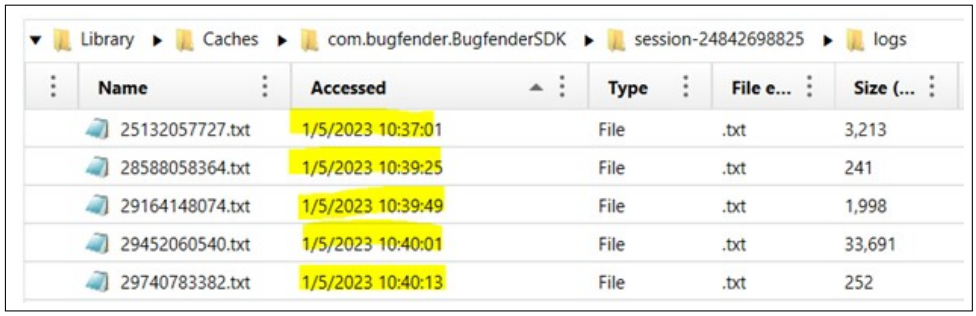


Figure 11. Logs created during the data population on the iOS device.

We were able to locate records associated with different events performed as part of the study. For example, Figure 12 presents the file `61708060319.txt` containing information that shows when the alarm was switched off, which also matched our timestamp of this event. Figure 13 shows the log related to the action of pairing the camera with the application by scanning the QR code on the camera device during the setup. Figure 14 shows the information such as uid, sid, wlanMac, and serial number presented in file `69196065763.txt`.

Figure 12. Action of switching to the OFF mode in the SimpliSafe app on the iOS device.

Figure 13. Process of scanning the QR code in order to pair the camera with the SimpliSafe app on the iOS device.

Figure 14. Various pieces of account information found on SimpliSafe app on the iOS device.

As it can be seen in Figure 15, we changed the state of the system from OFF to Home on 6 January 2023 at 10:17. The request did not show the initial state of the system; however, it showed the requested state, which in the previous image was Home. This was also consistent since the system was reverted back to OFF at 10:18 (see Figure 16).

Figure 15. Action of changing the state from OFF to Home in the SimpliSafe app on the iOS device.

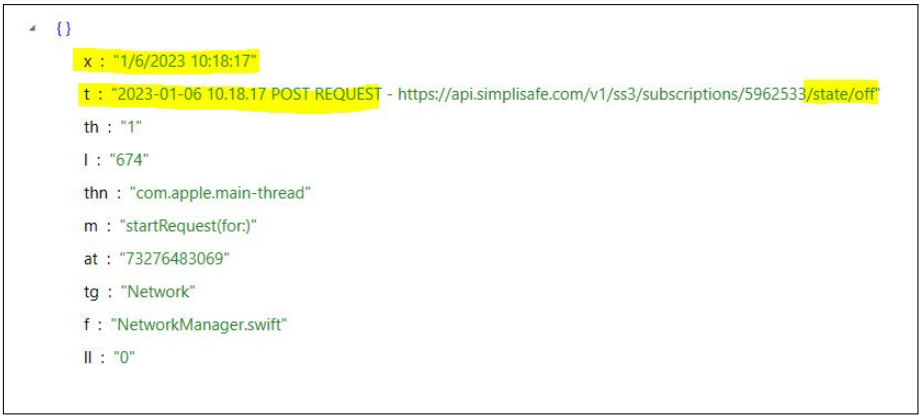


Figure 16. Action of changing the state from Home to OFF in the SimpliSafe app on the iOS device.

Figure 17 shows the system changing from OFF to Away with the 45-second countdown as recorded in file `75244046725.txt` at `\Library\Caches\com.bugfender.BugfenderSDK\session-24842698825\logs`.

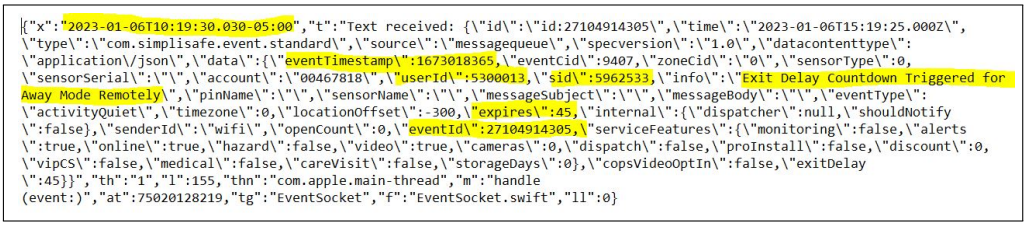


Figure 17. Action of changing the state from OFF to Away in the SimpliSafe app on the iOS device with a 45 s countdown.

Figure 18 shows the action of the camera being closed from the application logged in file `79276065553.txt`.

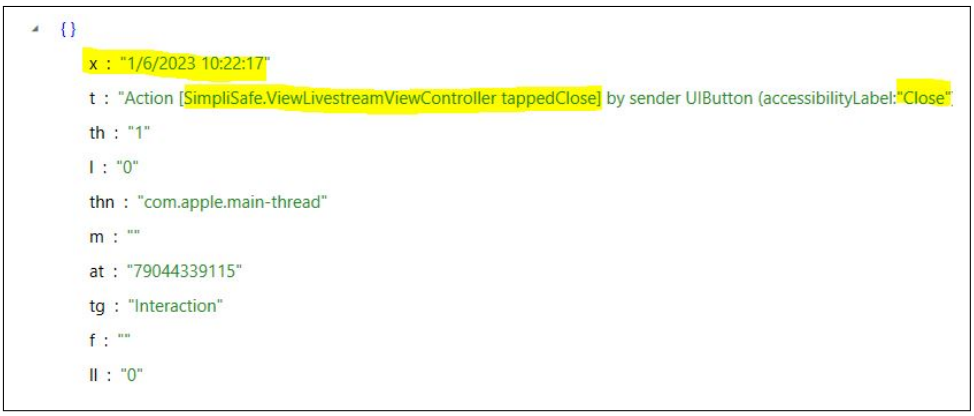


Figure 18. Action of closing the camera in the SimpliSafe app on the iOS device.

Figures 19 and 20 show the logs associated with the triggering of the front door sensor and the camera motion detection, respectively, as recorded in file `83020064349.txt`.

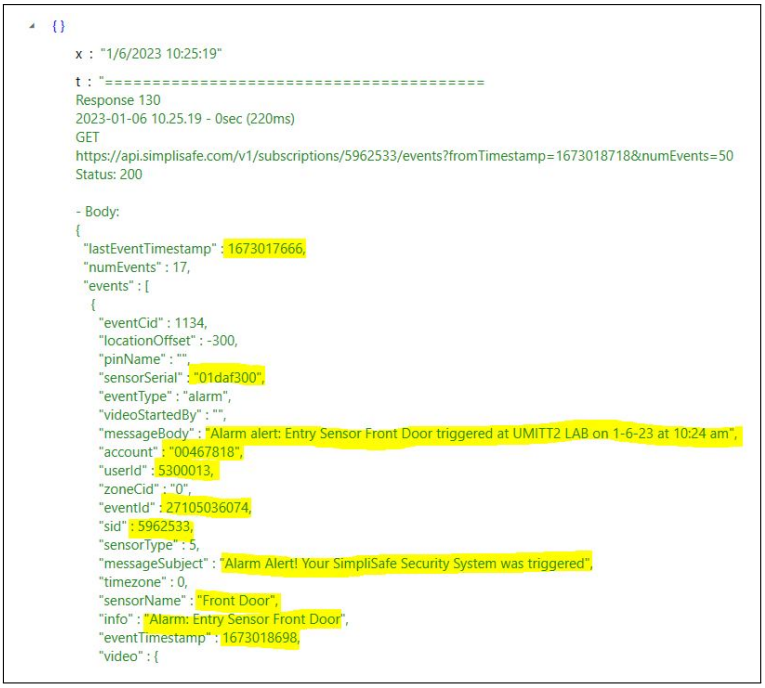


Figure 19. Log of triggering the front door entry sensor in the SimpliSafe app on the iOS device.

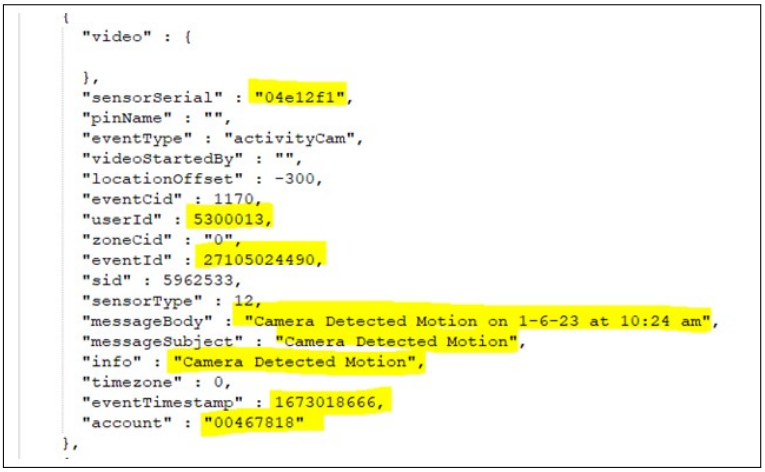


Figure 20. Log of the camera motion detection in the SimpliSafe app on the iOS device.

The SimpliSafe system has the ability to be disarmed remotely using the application. This feature was tested and we were able to extract the information shown in Figure 21.

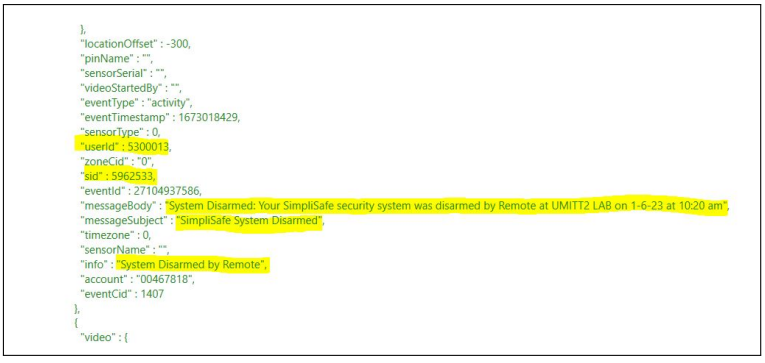


Figure 21. Log of remotely disabling the system by using the SimpliSafe app on the iOS device.

We found a thumbnail photo (Figure 22) stored at `\Library\Camera-Thumbnails`. To the best of our knowledge and based on the logs we kept of all the actions we performed, this thumbnail was not created as a result of any of our intentional actions. The metadata analysis performed on the photo showed three different times, accessed, modified, and changed. Accessed and modified times matched 6 January 2023 at 10:29 which was when the camera was first opened. The changed time was 6 January 2023 at 10:34, when the whole system was powered down leading us to believe that the camera took a thumbnail photo on the initial opening of the camera lens.

Figure 22. Thumbnail photo taken by camera found on SimpliSafe app on iOS device.

Finally, the email address associated with the SimpliSafe application was found in the `.plist` file at the location `FullFileSystem.1.dar\8E91EBEE-4046-4220-86EE-94CAA7DF32FD\Library\Preferences\com.simplisafe.mobile.plist` (see Figure 23).

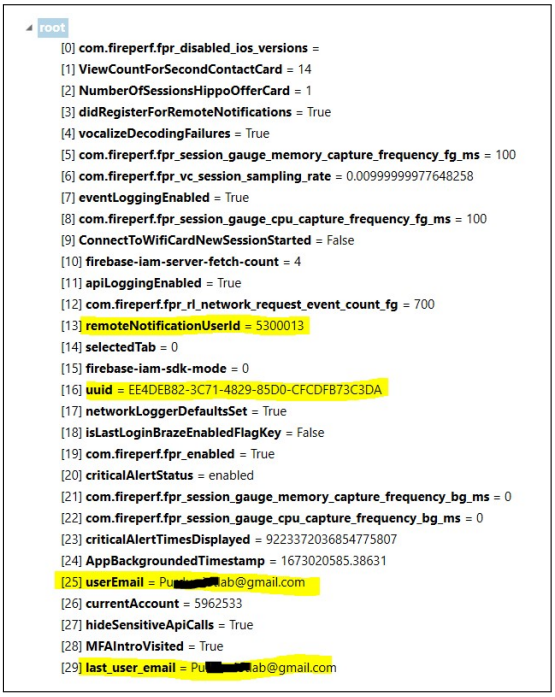


Figure 23. Email information found in the SimpliSafe app on the iOS device.

In summary, Table 3 summarizes the recovered artifacts from both platforms. In this table, "System Location" refers to any artifacts related to the physical location of the SimpliSafe system; "User Interactions" refer to any artifacts relating to user changes to the system, such as changing the system alarm state, viewing the camera feed, etc.; lastly, "App Usage" relates to any artifacts that can relate to how the user used the app or made setting changes, etc.

Table 3. Summary of recovered artifacts from Android and iOS

Artifact	Android	iOS
User's Name	No	Yes
User's Email	Yes	Yes
System Location	Yes	Yes
User Interactions	No	Yes
App Usage	Yes	Yes

4.3. Network Findings

The analysis of network captures did not show any unexpected or inexplainable patterns in the network traffic. Figure 24 shows the I/O graph from the first Wireshark capture. All spikes in network traffic were consistent with regular network traffic. TCP error packets increased only when the number of total transferred packets increased, which was consistent with normal traffic patterns.

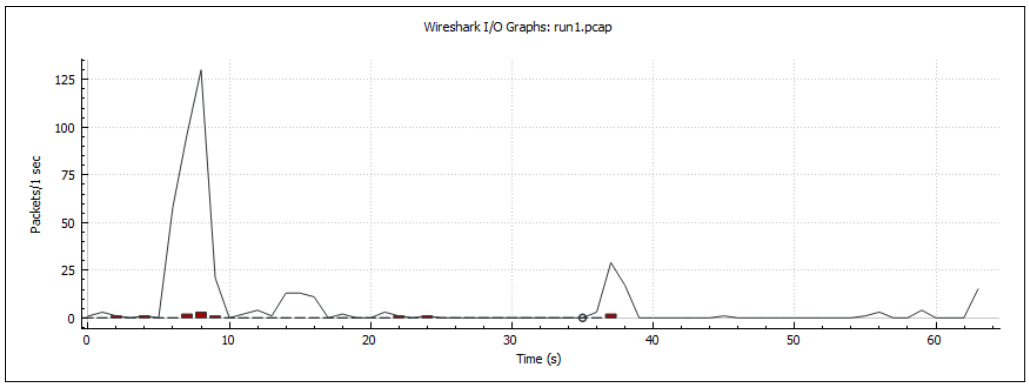


Figure 24. I/O Graph for the action of changing the state from OFF to Away in the SimpliSafe app.

As mentioned earlier, during the data acquisition phase, we changed the state from OFF to Away while capturing the network traffic and repeated this scenario ten times. Using PCAP Analyzer for Splunk, we were able to automatically generate various visual representations of the ten repeated packet captures and combine them onto a dashboard. Figure 25 shows a section of the PCAP Analyzer for the Splunk dashboard.

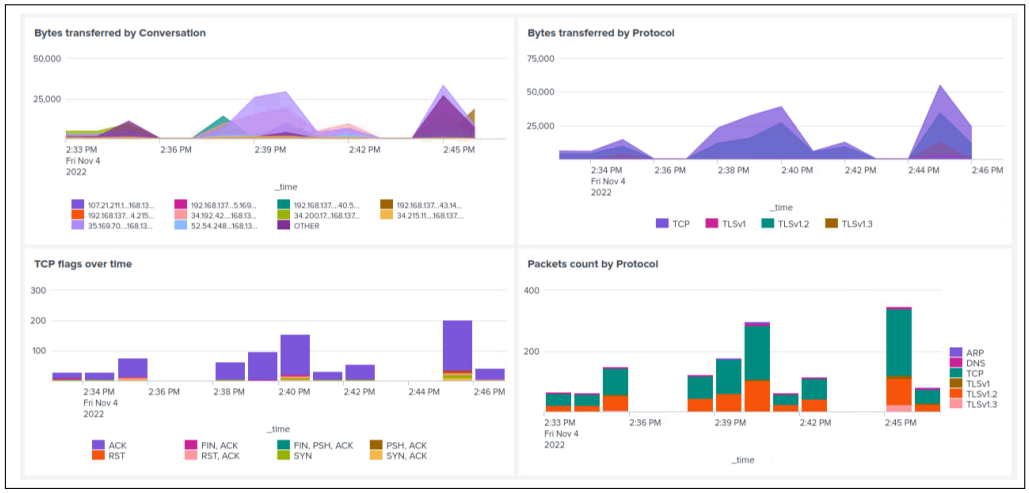


Figure 25. PCAP Analyzer for Splunk's dashboard showing network traffic graphs from OFF to Away packet captures repeated 10 times.

Based on the ten packet captures, there were a total of 1491 total packets transferred, with a packet loss of 3.89%. Regarding the bytes transferred by protocol and the packet

count by protocol, most bytes and packets were transferred using the TCP and TLSv1 protocols. The majority of the TCP flags over time were ACK flags. TCP errors over time increased and decreased, depending on the number of packets transferred, which was consistent with normal network traffic. Nothing unusual from the maximum round trip times and window sizes was detected.

The "tls && ip.src==192.168.137.141" was used to filter only Transport Layer Security (TLS) packets that came from the 192.168.137.141 source IP address (Android phone) to inspect the encrypted packets further. The Wireshark analysis showed that SimpliSafe was using TLSv1.2 to authenticate and encrypt data over the network. It was found that the key-exchange protocol used elliptic-curve Diffie–Hellman with a 65-bit public key. The Change Cipher Spec protocol was used to generate a session key and let the other party know that the message was encrypted. A 256-bit signature was used.

Given that SimpliSafe uses secure TLS, elliptic-curve cryptography, and best key-exchange practices, this may be a reason why most network traffic analysis findings were as expected. SimpliSafe follows the best practices for network security as determined by [22] when it comes to safeguarding its network traffic. As such, a substantial amount of the SimpliSafe network traffic that was analyzed was encrypted.

5. Discussion

Smart security systems continue to infiltrate homes at pace. However, with respect to the security and privacy of homeowners, these systems carry various risks. For instance, we were able to determine the time of entry and exit of the residents by analyzing the retrieved log records from an iPhone X. Should these log files hold data spanning multiple weeks, the homeowners' "pattern of life" could be determined. This insight could compromise the user's privacy and possibly pose a security risk. If such data were accessible by malicious parties, it would enable them to strategically plan and carry out heinous crimes, such as robbery, kidnapping, or murder.

The use of an indoor camera also adds to the potential for privacy concerns. Our findings from both operating systems indicated that it was possible to recover thumbnails from the camera. Such artifacts may capture the moment of interest when considering a criminal investigation.

Due to SimpliSafe's implementation of essential security standards and best practices, the majority of the collected network traffic was encrypted. This prevented the means of discovering what network traffic could be captured and what could not. No distinct patterns that could report behaviors related to various actions taken on the SimpliSafe mobile application were found. Although this hindered the investigation process in terms of network findings, it showed that the network artifacts that can be recovered from SimpliSafe devices are limited.

Challenges

The results of this research also highlighted numerous challenges that forensic examiners might face, particularly the disparities in the quantity of retrieved data available on different smartphones and operating systems. Moreover, it was found that the time between the data population and the device acquisition could have an impact on how many data could be retrieved. During this investigation, we used two different iOS devices, an iPhone SE and the iPhone X. At first, we used the iPhone SE with iOS version 14.0.1. After the data population, the image acquisition process involved Cellebrite UFED 4PC and Physical Analyzer acquiring the advanced logical image. Upon further examination and analysis of the created image, no significant data were found related to the SimpliSafe application and the activities carried out. The time between the data population and the data acquisition was more than a few days. The initial data population was conducted on 19 August 2022, and the acquisition was performed on 24 August 2022. This could be a possible explanation for the missing data. The second data population was performed on an iPhone X with iOS 14.2 (18B92). The process for the data population was very similar

to the first, with some minor variations. The main difference was the phone and the time between the data population and the acquisition, which was performed on the same day. The second method generated more results. Due to negligent relevant findings from the first acquisition of the iPhone SE, only the findings recovered from the iPhone X were reported in this research in Section 4.2 (iOS Findings). The data population process was the same for both devices, leading us to believe that the time between population and acquisition might be essential to the investigation. This hypothesis, however, needs to be tested further to prove the exact reason for the variation in the quantity of recoverable data.

During the data population phase, the SimpliSafe system needed to be unlinked from the Android account (email address) before it could be used with the Apple account. For this to happen, SimpliSafe requires customers to contact customer service to have them unlink the system on their end, a time-consuming process. Despite the inconvenience, this step adds another layer of protection for customers, since a stolen or second-hand preconfigured SimpliSafe system cannot be linked to a new account without having prior information about the previous owner.

6. Conclusions and Future Work

This paper addressed the need for transparency and increased user awareness concerning the use of IoT-enabled home security systems. Due to the lack of forensic research involving IoT-enabled security systems and the widespread use of these systems by American homeowners, the current work is uniquely positioned to both (1) document the forensically relevant artifacts available on Android and iOS smartphones for forensic investigators and (2) elucidate the privacy concerns that these systems' users may experience.

In this regard, we focused on self-installed home security systems, as their popularity is increasing. We performed a comprehensive forensic investigation of the SimpliSafe security system on both Android and iOS devices, and the system's network traffic by using state-of-the-art forensic software. The aim of our efforts was to detect any security and privacy concerns that arose when using such systems and to equip other investigators with a road map to study similar home security systems. Our findings highlighted the disparity in recoverable artifacts from various media. For instance, the actions the user performed on the app (e.g., changing the system state) were recovered from JSON and PList files on the iOS image, but such artifacts were not recoverable from the Android image. In terms of network traffic, SimpliSafe follows security standards and best practices according to [22]. The SimpliSafe system uses secure TLS, and elliptic-curve encryption, and properly generates session keys using the Change Cipher Spec protocol.

Future work for this project includes reconstructing the data population and taking images during different time intervals to determine the level of artifacts and their degradation over time. The results can be helpful for digital forensic investigators if there is limited time for the data acquisition process. Furthermore, the camera took a photo as a thumbnail, which can have an impact and reveal sensitive information. To better understand the time and anticipate when the camera is taking the thumbnail, we plan to place a clock in front of the camera showing the date and time. We also plan to further investigate the network traffic and recoverable artifacts while using the different subscription plans in which the system is being monitored remotely by SimpliSafe, where they can dispatch fire/police services when the alarm is triggered. Of course, this requires advanced planning and possibly SimpliSafe's cooperation in order to simulate a test environment without actually dispatching the emergency services.

Author Contributions: Conceptualization, S.H. (Shinelle Hutchinson), M.S., S.H. (Shiva Houshmand), and U.K.; methodology, S.H. (Shinelle Hutchinson), M.S., S.H. (Samuel Ho) and U.K.; validation, S.H. (Shinelle Hutchinson), M.S., S.H. (Samuel Ho) and U.K.; formal analysis, S.H. (Shinelle Hutchinson), M.S. and S.H. (Samuel Ho); investigation, S.H. (Shinelle Hutchinson), M.S. and S.H. (Samuel Ho); resources, S.H. (Shiva Houshmand) and U.K.; writing—original draft preparation, S.H. (Shinelle Hutchinson), M.S., S.H. (Samuel Ho) and S.H. (Shiva Houshmand); writing—review and editing, S.H. (Shinelle Hutchinson), M.S., S.H. (Samuel Ho), S.H. (Shiva Houshmand) and U.K.;

visualization, S.H. (Shinelle Hutchinson), M.S. and S.H. (Samuel Ho); supervision, S.H. (Shiva Houshmand) and U.K.; project administration, S.H. (Shiva Houshmand) and U.K.; funding acquisition, S.H. (Shiva Houshmand) and U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Appendix A.1. Full Timeline of the Events Performed on the Android Device (See Table A1)

Table A1. Timeline of events performed on the Android device while using the SimpliSafe app.

Log Date (mm/dd/yyyy)	Log Time (24 h, EST)	Action
12/12/2022	N/A	Setting up camera
12/12/2022	N/A	Allowed location access
12/12/2022	N/A	Allowed while using app
12/13/2022	10:16	Opened SimpliSafe app
12/13/2022	10:19	Pressed sign in
12/13/2022	10:19	Signed in successfully
12/13/2022	10:30	Clicked Refer+Edit
12/13/2022	10:30	Clicked no (output: unable to connect with the base station.)
12/13/2022	10:32	Base station connected to WiFi successfully
12/13/2022	10:36	Reset camera
12/13/2022	10:38	WiFi password incorrect
12/13/2022	10:38	Reentered WiFi password
12/13/2022	10:39	Camera connected to WiFi successfully
12/13/2022	10:41	Closed SimpliSafe phone app
12/13/2022	10:41	Opened SimpliSafe phone app
12/13/2022	10:42	OFF to Home
12/13/2022	10:43	Home to Away
12/13/2022	10:44	Opened front door sensor (output: there is a power outage)
12/13/2022	10:45	Away to Home
12/13/2022	10:46	Home to OFF
12/13/2022	10:46	Clicked no (output: help improve (star rating))
12/13/2022	10:49	Clicked 1 on keypad (output: connected to base station)
12/13/2022	10:49	OFF to Away via keypad
12/13/2022	10:50	Opened front door sensor
12/13/2022	10:52	Walked in front of motion sensor
12/13/2022	10:53	Alarm sounded
12/13/2022	10:53	Entered pin 1818 on keypad (Away to OFF)
12/13/2022	10:53	Received notification (alarm triggered at 101 Grant St)
12/13/2022	10:56	OFF to Away
12/13/2022	10:56	Triggered front door entry sensor
12/13/2022	10:56	Walked in front of camera
12/13/2022	10:58	Triggered front door entry sensor
12/13/2022	10:58	Heard SimpliSafe base station alarm ring
12/13/2022	10:58	Turned alarm off
12/13/2022	11:00	Viewed camera feed
12/13/2022	11:01	Walked in front of camera
12/13/2022	11:02	OFF to Away using keypad
12/13/2022	11:02	Opened front door sensor
12/13/2022	11:02	Walked in front of camera
12/13/2022	11:02	Saved safe word "Hotdog"
12/13/2022	11:03	Family exit home, OFF to Away via keypad
12/13/2022	11:03	Alarm on
12/13/2022	11:08	Camera triggered
12/13/2022	11:09	Basement triggered (basement door open and close)
12/13/2022	11:09	Motion sensor triggered
12/13/2022	11:09	Alarm sounded
12/13/2022	11:09	Received notification
12/13/2022	11:09	Turned off alarm via app

Appendix A.2. Full Timeline of the Events Performed on the iOS Device (See Table A2)

Table A2. Timeline of the events performed on the iOS device while using the SimpliSafe app.

Log Date (mm/dd/yyyy)	Log Time (24 h, EST)	Action
1/5/2023	10:19	iPhone X setup
1/5/2023	10:33	Added passcode 000000
1/5/2023	10:34	Downloaded the SimpliSafe app
1/5/2023	10:36	Opened app
1/5/2023	10:37	Signed in
1/5/2023	10:38	Used recovery code
1/5/2023	10:40	Accessed the app
1/6/2023	10:00	Set up the app
1/6/2023	10:08	Connected base to the WiFi
1/6/2023	10:09	Opened SimpliSafe app
1/6/2023	10:09	Switched from OFF to Home
1/6/2023	10:09	Allowed alerts for the app
1/6/2023	10:11	Reset camera settings through the app
1/6/2023	10:14	Attempted to connect to the WiFi
1/6/2023	10:15	Connection successful
1/6/2023	10:15	Tested camera On/Off
Following Actions were all performed from the SimpliSafe Application		
1/6/2023	10:17	OFF to Home
1/6/2023	10:18	Home to OFF
1/6/2023	10:19	OFF to Away (including 45 s countdown)
1/6/2023	10:20	Away to OFF
1/6/2023	10:21	Opened camera (watch live)
1/6/2023	10:22	Closed camera
The following actions were triggering alarms with various sensors		
1/6/2023	10:23	OFF to Away (including 45 s countdown)
1/6/2023	10:24	Triggered front door sensor
1/6/2023	10:24	Triggered camera
1/6/2023	10:25	Alarm sounded
1/6/2023	10:25	Alarm turned off from the app
Another action		
1/6/2023	10:26	OFF to Away (including 45 s countdown)
1/6/2023	10:17	Triggered camera
1/6/2023	10:28	Basement motion sensor triggered
1/6/2023	10:28	Alarm sounded
1/6/2023	10:28	Alarm turned off from the app
Another action		
1/6/2023	10:29	Opened Camera
1/6/2023	10:29	Talked into camera from the app
1/6/2023	10:30	Camera off
Another action		
1/6/2023	10:30	OFF to Away (including 45 s countdown)
1/6/2023	10:32	Front door sensor activated
1/6/2023	10:32	Basement Motion Sensor triggered
1/6/2023	10:32	Camera triggered
1/6/2023	10:32	Alarm sounded
1/6/2023	10:33	Alarm turned off from the app
1/6/2023	10:34	Powered off everything

Note: Prior to the imaging process the application was not cleared out of the memory and the imaging was performed right after.

References

1. Research. Markets. Home Security Systems Market by Home Type. Available online: https://www.researchandmarkets.com/reports/5130165/home-security-systems-market-by-home-type (accessed on 10 July 2022).
2. Staff, G. Alexa, Did He Do It? Smart Device Could Be Witness in Suspicious Florida Death. Available online: https://www.theguardian.com/us-news/2019/nov/01/alexa-florida-death-witness-amazon-echo (accessed on 10 July 2022).
3. Whittaker, Z. Judge Orders Amazon to Turn Over Echo Recordings in Double Murder Case. Available online: https://techcrunch.com/2018/11/14/amazon-echo-recordings-judge-murder-case/ (accessed on 20 January 2023).
4. Home Security Solution Market by Type. Available online: https://www.researchdive.com/293/home-security-solutions-market (accessed on 20 March 2023).
5. SimpliSafe Home Security. Available online: https://www.simplisafe.com (accessed on 10 July 2022).
6. Priest, D.; Anders, D. Best DIY Home Security Systems for 2022. Available online: https://www.cnet.com/home/security/best-diy-home-security-systems/ (accessed on 10 July 2022).

7. Hutchinson, S.; Yoon, Y.; Shantaram, N.; Karabiyik, U. Internet of Things Forensics in Smart Homes: Design, Implementation, and Analysis of Smart Home Laboratory. In Proceedings of the 2020 ASEE Virtual Annual Conference, Virtual, 22–26 June 2020. [CrossRef]
8. Chung, H.; Park, J.; Lee, S. Digital Forensic Approaches for Amazon Alexa Ecosystem. *Digit. Investig.* **2017**, *22*, 15–25. [CrossRef]
9. Dorai, G.; Houshmand, S.; Baggili, I. 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, Hamburg, Germany, 27–30 August 2018.
10. Hutchinson, S.; Karabiyik, U. Forensic Analysis of the August Smart Device Ecosystem. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; pp. 1–7.
11. Janes, B.; Crawford, H.; OConnor, T. Never Ending Story: Authentication and Access Control Design Flaws in Shared IoT Devices. In Proceedings of the 2020 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 21 May 2020; pp. 104–109. [CrossRef]
12. Connor, T.O.; Jessee, D.; Campos, D. Through the Spyglass: Towards IoT Companion App Man-in-the-Middle Attacks. In Proceedings of the Cyber Security Experimentation and Test Workshop. Association for Computing Machinery, Virtual, 9 August 2021; pp. 58–62.
13. Ndatinya, V.; Xiao, Z.; Manepalli, V.R.; Meng, K.; Xiao, Y. Network forensics analysis using Wireshark. *Int. J. Secur. Netw.* **2015**, *10*, 91–106. [CrossRef]
14. Ho, S.; Greeson, H.; Karabiyik, U. Smart Home Forensics: Identifying Ddos Attack Patterns on Iot Devices. In Proceeding of the 2022 ADFSL Conference on Digital Forensics, Security and Law, Virtual, 25–26 July 2022; pp. 1–12.
15. Lyon, G. Nmap Security Scaner. Available online: https://https://nmap.org/ (accessed on 3 January 2023).
16. National Institute of Standards and Technology (NIST). Available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-202.pdf (accessed on 23 January 2023).
17. Cellebrite UFED 4PC. Available online: https://cellebrite.com/en/ufed/ (accessed on 3 February 2023).
18. Magnet AXIOM Forensics. Available online: https://www.magnetforensics.com/products/magnet-axiom (accessed on 3 February 2023).
19. Sharpe, R.; Warnicke, E. Wireshark User's Guide. 2011. Available online: https://www.wireshark.org/docs/wsug_html_chunked/index.html. (accessed on 1 March 2022).
20. Splunk. Available online: https://www.splunk.com/ (accessed on 3 February 2023).
21. Schwartz, D. PCAP Analyzer for Splunk. Available online: https://splunkbase.splunk.com/app/2748 (accessed on 3 February 2023).
22. Kizza, J.M. *Computer Network Security*; Springer: Berlin/Heidelberg, Germany, 2005.