Welcome Pwn - Almond Smart Home Hub Forensics

By

Akshay Awasthi, Huw O.L. Read, Konstantinos Xynos, Iain Sutherland

From the proceedings of
The Digital Forensic Research Conference
DFRWS 2018 USA
Providence, RI (July 15th - 18th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment.

As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and forensic challenges to help drive the direction of research and development.

https://dfrws.org
Many home interactive sensors and networked devices are being branded as “Internet of Things” or IoT devices. Such disparate gadgets often have little in common other than that they all communicate using similar protocols. The emergence of devices known as “smart home hubs” allow for such hardware to be controlled by non-technical users providing inexpensive home security and other home automation functions. To the cyber analyst, these smart environments can be a boon to digital forensics; information such as interactions with the devices, sensors registering motion, temperature or moisture levels in different rooms, all tend to be collected in one central location rather than separate ones. This paper presents the research work conducted on one such smart home hub environment, the SecuriFi Almond, and provides guidance for forensic data acquisition and analysis of artefacts pertaining to user interaction across the hub, the iPhone/Android companion applications and the local & cloud-based web interfaces.

© 2018 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/).
wireless network transmission protocols. This is now a significant area of concern with research efforts focused on the analysis and data extraction from popular IoT devices, e.g., Meffert et al. (2017) and Oriwoh et al. (2013). The complexity, variety and distribution of IoT devices, which are by their nature part of an infrastructure, may cause significant problems. Simply gaining physical access to the systems can be a separate challenge altogether for the analyst. However, in many cases, there is little of forensic value on the devices themselves. What may prove to be of greater importance is accessing any system used to integrate IoT devices providing centralised control (Sutherland et al., 2015). Typically, domestic systems are connected via some form of hub or central service to facilitate a “Smart Home”. The integration of these devices has already raised security concerns (Plachkinova et al., 2016). Generally, the forensic analysis of Smart Hubs thus far has been limited. The following section describes related work carried out to date on three such systems; Amazon Alexa, Apple HomeKit and Google OnHub or Google Home.

### 2.1. Amazon Alexa

The Amazon Alexa system is a combination of specific hardware (Echo and Echo Dot) and the cloud based Alexa personal assistant. The considerable popularity of the Amazon Alexa System has led to some community efforts exploring the analysis of the device including the hardware (dj_skully, 2016). An analysis by the LCDI (2016) provided some insight regarding performing a forensic analysis on the Amazon Alexa, via third party devices. The report explains techniques for data collection and data extraction. The greatest challenge they encountered was third-party device integration with the Echo. The data collection using such devices and their companion applications was found to be generating possible discrepancies in the data. Chung et al. (2017) considered the Alexa ecosystem and proposed a possible toolkit to support forensic analysis; it tries to acquire (download) cloud-native artifacts from the server using the unofficial APIs…. A challenge experienced by the authors in the past is unofficial APIs are subject to change without warning which could then require revising of code, that is if the functionality is still available. Hyde and Moran (2017) describe both destructive and non-destructive methods of accessing the Amazon hardware to extract evidence.

### 2.2. Apple HomeKit

The Apple system uses the iCloud keychain to retain information on devices and other information and requires an Apple iOS device or Apple TV to remain in the home to act as a hub for external access (Apple, 2017a). Apple released the HomePod in early 2018, which appears to be limited in capacity acting as a speaker and an interface to Siri and HomeKit devices (Apple, 2017b). Given Apple’s public stance on encryption and working with law enforcement (Cook, 2016), the challenge of extracting forensic data from the Home environment will likely be of particular interest to digital forensic researchers.

### 2.3. Google OnHub and Google Home

Google Home provides a similar service to that of Alexa with access to various Google services and Google assistant. It is capable of running on either the Android or Apple iOS Operating Systems. Launched in 2017, it can interface with a number of IoT devices, there is however very limited information on forensic best practice with this system. Another possible device the investigator might encounter is the Google OnHub (Google, 2017) which takes a different approach than that adopted by Amazon. Rather than becoming an additional device on the network, the OnHub is intended to replace the home router with one system that can interface with Smart/IoT devices.

### 3. Almond ecosystem

The Almond+ is a smart home hub that integrates the functionality of a router with the ability to control and respond to IoT sensors and devices. It has the ability to work with or without Internet connectivity (Securifi, 2017). It is more akin to devices such as Googles OnHub, than Amazon’s Echo, in that it is designed to replace an existing router. The device also provides the facility to be setup as a repeater or an access point. The Almond + supports two IoT protocols, namely Zigbee and Z-Wave. Fig. 1 demonstrates how the following sensors were connected to the Almond + environment for the experiment:

- Three Philips Hue Lamps via a Philips Hue Bridge - changes colour, dimming, on/off
- Jasco Dimmer Plug, 3-pronged dimmer device
- Securifi Peanut Plug, on/off power device
- Fibaro Door/Temperature Sensor, two-components
- NYCE Motion/Temperature/Humidity Sensor, positioned on ceiling
- Two NYCE Door Sensors, alternative to Fibaro

There are four ways for a user to interact with the Almond ecosystem, via the hardware itself, a companion app on iOS or Android, and through Cloud or local web interfaces.

#### 3.1. Via the touchscreen

The Almond+ provides an interactive touch screen to the user as depicted in Fig. 2. The interface on the Almond + provides a myriad of information to the user: settings, adding and controlling sensors, weather, list of users, firewall, security and sharing features.

#### 3.2. Via the companion app

The smart app for the Almond+ is available for iOS and Android (Fig. 2) and both have a consistent look and feel. The app provides more information than the web interface and is able to connect locally (LAN) or via cloud to the Almond + router. In local mode, the environment does not need Internet connectivity to work. The Cloud connectivity feature provides the user facility to monitor and control the smart sensors connected to the Almond + remotely. The cloud connectivity also provides the history for the sensor activity.

![Fig. 1. Almond+ Environment Setup.](image-url)
It sends a notification when a sensor is triggered and its current state. It also maintains a log of the sensors activity in the app. We found only subtle differences between the companion app for iOS and Android; the structure of the SQL databases were slightly different though the data was largely the same (Fig. 4).

3.3. Via the cloud-based interface

The web interface appears to not be as complete as the Almond + touchscreen software or companion apps. This provides a view of the current status of sensors and which network devices have connected (Fig. 2).

3.4. Via the local web interface

The hub provides a local web-based interface which may be accessed by visiting the IP address on the local network (in our setup this was achieved by visiting http://10.10.10.254). It is similar in look and feel to the Cloud web interface, but is only accessible on the local connection.
4. Forensic challenges

During the course of the research, a number of barriers were identified that could impede a successful forensic examination. No discernible removable media could be found in any of the hardware in the sample environment, neither in the IoT devices themselves nor in the smart hub. The Almond+ contains 512 MB of RAM and flash memory. The flash memory tends to be used for core operating system files; a temporary files system stored in volatile memory contains sensor data. It is of critical importance to note that, if the smart hub is rebooted or powered off, valuable digital evidence will be irretrievably lost. As may be seen in Fig. 3, almost half of the available RAM is dedicated to /tmp which contains a considerable number of artefacts (211.6M of 512M total).

A popular mechanism for accessing remote systems is SSH, which is enabled by default on the cabled LAN connections; user configuration is required to enable it over Wi-Fi (Lars, 2014a). SSH can permit remote access, filesystem extraction (sftp) and, in combination with other tools like dd, remote imaging is also possible. An investigation into the potential for SSH as a data-access and data-extraction mechanism will be explored later in the paper.

There are several devices supported by the Almond+ that can also be controlled by other “smart” devices; for example, the Philips Hue lightbulbs are compatible with the Amazon Echo device. Understanding whether there is any synchronisation between different controller-devices will be of interest to the forensic examiner.

Unfortunately for the examiner, it has become more difficult to obtain data from Apple devices in recent iOS versions with fewer files from applications included in iTunes backups for offline analysis. The iTunes sync functionality is so important to iOS forensics, even a clean install of Cellebrite Physical Analyzer v.6.3.2 states that iTunes must be installed for extraction from certain iOS devices. The reader is urged to refer to the summary of iOS-11 in Afonin (2017) for general iOS challenges to forensic analysis. Significant files from the companion app are not easily accessible in iOS; analysts may need to either manually hand-scroll through information on the app itself or risk jailbreaking the device to obtain the data. Similarly, depending upon the state of forensic tools, the Android version of the Almond app may have extraction issues related to permissions.

5. Investigative methods

In order to create the IoT-enabled environment, a number of sequential steps were completed to assess the impact of different devices (IoT, other controller-devices) and whether any artefacts of importance to forensic examiners could be discovered.

1. Initial assessments of Almond + logs via an empirical investigation
   - IoT device introduced to Almond + ecosystem
   - Home hub log files examined and observations recorded
   - IoT device removed, to reduce noise on network

2. Almond + touchscreen assessment
   - Update configuration for an IoT device
   - Record observations in Almond + logs

3. Companion app assessment
   - Install Almond by Securiapp on iPhone and Android
   - Interact with devices via app, note time
   - Interact with devices via other mechanisms e.g. Cloud
   - Record observations in Almond + logs and handsets

4. Local network and Cloud assessment

5. Alternative controller-device implications (Amazon Echo)

Post Almond + setup, the smart devices were added sequentially. Sensors were individually connected to the Almond + to see what changes were being made in the log files. After examining the modifications made by a sensor the device was removed from the Almond+ and the router was setup for the next sensor. This allowed us to identify generic logs which stored all actions, and specific logs unique to certain devices.

After recording the readings for each sensor individually the Almond+ was reset and all the sensors were connected to it. Then the logs were monitored to see what was being recorded on the
Almond + when the sensors were interacted with via the on-screen interface, the smart app and the Cloud/web interface. The logs were cross referenced to observe how, if any, changes occurred and to determine the results of the users actions.

Amazon Echo device was used on the same network to control the Philips Hue lights that were also connected to the Almond+. The logs were examined to evaluate if either any purposeful synchronisation took place between controller-devices, or if any such device was able to observe the controlling instructions provided by another.

NIST 800-101 Guidelines on Mobile Device Forensics (Ayers et al., 2014) was referenced before attempting the following course of action. We assessed the implications of performing a Pangu jailbreak (Pangu, 2016) on an iPhone 5s running iOS 9.3.2. Comparisons were made between data extracted from 1) a backup made using iTunes with a non-jailbroken device, 2) a backup made using iTunes with a jailbroken device, 3) manual extraction using Cydia file browsing apps and 4) Cellebrite UFED Touch2 filesystem extraction. We found there wasn’t a difference in the quality of the extraction between 1, 2, and 4. but 3, provided additional files of importance. As we had the ability to jailbreak and were able to confirm that Pangu on an iPhone 5s running 9.3.2 did not alter the companion app files, we chose to proceed with this method to obtain more data. Other jailbreak tools should be assessed in a similar function to confirm the implications of the jailbreak process upon the quality of the extraction.

A similar process was conducted on the Android device (a Sony Xperia Z1 Compact running Android 5.1.1) extraction from stock 5.1.1 via the android debug bridge (adb), 2) the backup feature from the Sony Xperia Companion software (1.9.2), 3) extraction after rooting, 4) using forensic hardware (Cellebrite UFED Touch 2, firmware 6.1.2), Only methods 3 and 4 allowed the extraction of the filesystem. Of these, method 4 was used as it is an established method of data extraction.

6. Data extraction

There are stark differences between extraction methods for devices in the Almond ecosystem, each is discussed in turn below.

6.1. Almond + smart home hub

According to Lars (2014b), default SSH access is only available via LAN. At the time of writing, the firmware revision is AP2-R090-L009-W016-ZW016-ZB005. We have identified that whether SSH is enabled or disabled in the interface, SSH continues to be accessible via Wi-Fi. Enabling the repeater mode functionality greys out all the WAN access settings, but SSH via Wi-Fi continued to work. The only way we have been able to remove SSH access was to connect via SSH and kill the SSH process manually.

The SSH server is provided by Dropbear v0.52, released in 2008 (Johnson, 2017) and with 3 known vulnerabilities. When connected to a modern SSH client, the client responds that it is unable to negotiate a secure connection with the host; no matching key-exchange methods are found. Dropbear offers diffe-hellman-group1-sha1 which requires forcing legacy options to connect, guidance is available in OpenSSH. We had to use the following command to login to the hub with a default username and password of root/root:

```
ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 root@<IP>
```

After gaining access to the file system we proceeded with an empirical investigation of the filesystem for artefacts that would be of use to an investigation. This process led to identifying `/tmp` as containing most files of significance. The Almond + supports USB 3.0. We used a USB drive formatted appropriately to transfer files. The bitstream and resulting captures were hashed using the built-in `md5sum` to ensure standard forensic practice was followed. The following command was used to make the capture:

```
tar-czvf -/tmp/ | tee
/mnt/usb/id/time_date.tar.gz | md5sum >
/mnt/usb/id/time_date.tar.gz.md5
```

Furthermore, it is possible to forensically image the raw devices in a similar fashion. Using the following command, we were able to pull the flash memory directly via ssh and create hashes of the bitstream and resulting image file.

```
ssh root@<ip_address>
"dd if = /dev/mtdblock?" | tee block7.dd
| md5sum > block7.dd.md5
```

Given the system is active, imaging the flash memory twice will likely result in different hashes. Therefore, the importance of hashing the bitstream and the resulting image simultaneously is of paramount importance. If the md5 of the dd image and the text file match, the image has been created correctly.

We identified 14 separate mtdblock devices on the Almond+ and used the Linux tools `file & binwalk` to attempt identification. A summary of these may be seen in Table 1.

The squashes file system can be easily extracted by using squashfs tools, which in Ubuntu can be found in the package squashfs-tools.

Attempts to extract or mount the individual mtdblock images identified as JFFS2 were not particularly successful; the mounting process generated the following error: `jffs2: compression type 0x08 not available.`

This can be seen by attempting to mount the drive with the following set of commands:

```
$modprobe -r mtdram
$modprobe mtdram total_size = 18000
$dd if = block7.dd of = /dev/mtdblock0
$mount -t jffs2/dev/mtdblock0/mnt/testDrive
$mdmesg | tail
```

First, we ensure that mtdram is unloaded and then load it again with the correct size of the device when executing `modprobe`. This requires the analyst to check the file size of the image that will be loaded (e.g., block7.dd), this is then provided as a value in `total_size`. The image file is then dd’ed to the device (e.g.,/dev/mtdblock0) and it is then mounted. `mdmesg` can be consulted for any messages or errors.

It is hypothesised that the compression type (i.e., LZARI, type 0x08) is one that is not supported by modern implementations of jffs2 and was only included in the kernel of this particular system. There are a number of things that a forensic analyst could do. One of these could be to attempt to boot up the system using QEMU and navigate it from there. Another is to find the implementation of the compression and decompression used and implement it in `jffs2dump`.

6.2. Companion app

As previously mentioned in Forensic challenges, obtaining raw files from Apple devices is becoming progressively difficult with each iOS iteration. The process of hand-scrolling with suitable video recording equipment, albeit tedious, will extract the same useful information as can be seen in the raw files obtained via the jailbreaking method.
The tools iFile and Filza were used on an iPhone 5s jailbroken with Pangu (2016) on iOS 9.3.2 to locate and extract the Almond + companion app data for further analysis. Filza conveniently provides the human-readable names of apps located in the iPhone.

Retrieving data from the Android device (a Sony Xperia Z1 Compact running Android 5.1.1) was more straightforward; a Cel-ebrite UFED Touch 2 (firmware 6.1.0.140) provided a complete file system extraction including the Almond + app data. Once the Almond + folder is located, it may be copied to a workstation for analysis of the SQLite files. All such files were analysed using DB Browser for SQLite v.3.10.1 (sqlitebrowser, 2017).

6.3. Web interfaces – cloud and local

The Cloud web interface for the Almond + may be accessed at connect.securifi.com. An analyst may take screenshots or save the webpage to disk to extract information. It is not recommended to interact with the local web interface, as can be seen in section 7, mere browsing can cause updates to several files of evidential significance.

7. Artefacts of forensic importance

Across the Almond + environment, several areas can provide evidence of value to the forensic investigator. Depending on the interface, varying levels of detail may be obtained. Table 2 summarises important locations found across the Almond + and its companion applications.

7.1. Residing on the Almond +

After performing an extraction as detailed in the data extraction section of this paper, the following files were identified as containing artefacts of forensic significance.

/tmp/CloudDaemon.log - Provides a detailed history of data being sent to Cloud storage. All actions may be categorised by their CommandType, which appears as both XML and JSON entries. The following types were identified as being of particular interest:
- 24 and 25- Contain Wi-Fi SSID and password in plaintext
- DynamicDeviceList - Smart devices and sensor values
- DynamicSceneList - Scenes assigned to smart devices
- DynamicClientList - Network devices, whether or not active
- DynamicIndexUpdated - Smart device state changes
- RouterSummary - Router settings, uptime, SSID, IP
- GetWirelessSettings - SSID, mode, channel, encryption type, location

AlmondProperties Centigrade usage, URL, preferred Internet check URL

/tmp/autoip.json - Contains geographical information about the Almond + and the local weather summary. Information obtained via geographical lookup of IP address therefore VPN usage may provide weather patterns from different physical locations. Available information includes: timezone, city, state, country, longitude, latitude, weather conditions.

Finally, there are other files worth analysing depending on the smart sensors in use. For example, given its name, /tmp/NestServer.log is likely to contain details about events generated by Nest thermostats; we did not have a sample device at the time of writing to confirm residual artefacts.

7.2. Residing on the companion app

Common to both Android and iOS, the app provides two methods of interfacing with Almond, via Cloud connectivity or through local (LAN) connectivity. The local connectivity option provides similar features when connected via the Cloud but it neither records device history nor sensor actions.

As discussed in Zdzierski (2008), "[the iPhone makes heavy use of database files to store information]". Though the reference is nearly a decade old at the time of writing, the statement is relevant considering the plethora of apps storing data in SQLite database formats, on both Google and Apple operating systems. It should be reiterated that the iOS files retrieved from an iPhone were done so by jailbreaking the device as described in the Investigative methodology. The Almond + companion app, “Almond by Securifi” in both the Apple app and Google play stores, contains the following files of significance:

<root>/Documents/toolkit_devicelogs.db

iOS - Network devices (e.g. tablets, computers) are logged joining and leaving the network. With reference to Fig. 4a, two timestamps are available per event, both stored as GMT. date_bucket indicates the date (time is always set at midnight) and time which provides the full date and time. The indexvalue states when a new networked device was connected, <device_name> just joined Almond-aab8 (not seen in Fig. 4a), rejoined, or left. The devicename appears to be an amalgamation of both the device’s name and a copy of the indexvalue. It is unknown if this repetition is deliberate.

<root>/Documents/toolkit_notifications.db

iOS - State changes for smart devices (i.e. IoT) are recorded in this file. The database structure is the same as that observed in toolkit_devicelogs.db. Timestamps are again stored in GMT. devicename contains either a default or a value set by the user identifying the IoT devices by name. indexvalue contains quite a varied set of values, based on the devicetype. The value_indexname is populated with an indicator of what state change triggered the event. For example in Fig. 4b, devicetype 60, a motion sensor with a built-in humidity sensor, has two
Table 2
Summary of forensically-important locations.

<table>
<thead>
<tr>
<th>Source</th>
<th>Location</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Securiﬁ Almond+</td>
<td>/tmp/connected_home.log</td>
<td>Entries created when smart devices are used on Almond+</td>
</tr>
<tr>
<td>Securiﬁ Almond+</td>
<td>/tmp/association.log</td>
<td>Identified when smart device is added to hub</td>
</tr>
<tr>
<td>Securiﬁ Almond+</td>
<td>/tmp/CloudDaemon.log</td>
<td>Detailed log of data sent/received from Cloud</td>
</tr>
<tr>
<td>Securiﬁ Almond+</td>
<td>/tmp/autoip.json</td>
<td>Almond + geographical information and weather data</td>
</tr>
<tr>
<td>iPhone App</td>
<td>&lt;root&gt;/Documents/tool-kit_deviceLogs.db</td>
<td>Record of all network devices (dis)associating with Almond+</td>
</tr>
<tr>
<td>iPhone App</td>
<td>&lt;root&gt;/Documents/tool-kit_notification.db</td>
<td>Record of all smart devices which have alerts explicitly set</td>
</tr>
<tr>
<td>iPhone App</td>
<td>&lt;root&gt;;/Library/Caches/Snapshots/com.securif.almond/*82x.png</td>
<td>Screenshot of most recent user-interaction in app</td>
</tr>
<tr>
<td>Android App</td>
<td>/data/com.securi-fl.almondplus/</td>
<td>Record of smart devices which have alerts explicitly set and</td>
</tr>
<tr>
<td></td>
<td>databases/notiﬁcations.db</td>
<td>network devices (dis)associating with Almond+</td>
</tr>
<tr>
<td>Cloud/Web</td>
<td>connect.securiﬁ.com</td>
<td>Current Wi-Fi settings, list of all networked devices</td>
</tr>
</tbody>
</table>

value_indexname entries associated with it, HUMIDITY and a NULL entry. The indexvalue for HUMIDITY provides the percentage value when the humidity in the room changes, the NULL entry indicates whether motion has been detected (true/false values).

<root>/Library/Caches/Snapshots/com.securiﬁ.almond/*82x.png (* represents an 8-4-4-4-12 alphanumeric string) iOS - This image file provides a snapshot of whatever the user last viewed on the application. Both the content and the ﬁlename are updated whenever the application loses focus.

<root>/databases/notiﬁcations.db

Android - An amalgamation of both iOS SQLite databases compiled in one ﬁle. Though the database contains the same information as its iOS counterpart, it is structured differently. Fig. 4c has a single _time_stamp value in GMT which provides full date and time information (unlike data_bucket above). The _index_name, _name and _index_value are equivalencies of the following iOS values value_indexname, devicename and indexvalue. The value_message appears unique to the Android app, it provides a high-level textual description of the event. The information can be derived from using other values but could be for convenience. For example, row 39 in Fig. 4c shows a dimmer plug registered a SWITCH MULTILEVEL value of 37. The _message indicates dimmer plug is dimmed to 37%.

It should be noted that both the Android and iOS companion apps contain a “View Device History” option for both smart (i.e. Zigbee and Z-Wave) and network (i.e. PC with Wi-Fi) devices. The available “history” for smart sensors is obtainable via the Alerts timeline. This information appears to be streamed from the Cloud as it did not appear to exist in local database ﬁles. However, when the companion app is reinstalled and an existing ID is used, this prior history can be streamed from the Cloud service and may be seen again.

7.3. Residing on the cloud interface

At the time of writing, this interface is neither as developed as the companion app nor the native Almond + interface; both the “Rules” and “Advanced Features” provide messages stating that these features are unavailable. We were able to retrieve individual sensor data, presented as a timeline and a list containing current wireless settings, network devices (name, Mac and IP only) and the software version. The “View Device History” option seen in the companion app was absent, but the list does indicate which devices are currently connected to the hub.

7.4. Observations with the Amazon Echo

As mentioned in Forensic challenges, some devices can be controlled by multiple smart hubs. With an Amazon Echo device connected to the sample environment the authors were able to control the Hue lightbulbs with voice commands. Even though the Hue bridge was connected as a smart device to the Almond+ (via ethernet), any changes made to the state of the Hue lights with the Echo did not result in any artefacts generated on the Almond ecosystem. Only changes made either directly on the Almond + or via the companion app while in Cloud connectivity mode resulted in observational evidence on the Almond + device. The companion app in local mode provided current status and interaction, but no update to the logs. The companion smartphone app activity can be found on the Almond+ in the CloudDaemon.log ﬁle which maintains the records for the commands sent via the app. The work by Chung et al. (2017), should be referenced if an Amazon Echo is found in the target environment for further extraction opportunities.

8. Recommendations for the forensic examiner

The following suggestions are made to those who encounter such devices during an examination:

1. Investigate the Almond + ﬁrst; it is the most volatile. Do not reboot or reset, evidence resides in a volatile tmpfs type filesystem.
2. Do not touch the screen. Investigators may be tempted to view the “Connected Sensors” but this will generate events in /tmp/connected_home.log
3. Connect to Wi-Fi (if password available) or LAN on Almond+.
5. Create a backup of the data as described in section 6 a memory stick and hash the extracted data.
6. Analyse ﬁles highlighted in Table 2 from Almond + according to section 7.

If the Almond+ companion app is available on a phone/tablet, there are two paths an analyst may traverse but both should be performed only after standard procedures (Ayers et al., 2014) are considered:

1. Ensure device data is copied using standard operating procedures
2. Assess ability of forensic imaging tools (the authors used Cellbrite’s UFED Touch 2 for the Android device) to extract data from device.
3. If device is iOS-based and jailbreaking is an option, evaluate impact of jailbreak to be able to defend actions in a court of law
4. Install Filza or iFile from the Cydia App Store onto device, manually extract data described in section 6.
5. Analyse ﬁles highlighted in Table 2.
6. If jailbreaking (iOS) or rooting (Android) are not an option, start video recorder and begin capture.

7. Perform hand-scroll of Almond+ companion app paying particular attention to any Alerts - once viewed their status will change to "seen".
   a. Bell icon - provides timeline of alerts (for those explicitly setup by user)
   b. Devices icon - "View History" provides presence-based (i.e. within range) artefacts (note - only useful with Cloud connectivity enabled)

If the Cloud password is known:
1. Login at https://connect.securifi.com
2. Obtain router and device sensor via print screen and/or saving web pages.

9. Conclusions and future work

The research work presented in this paper focused on one of the many different hubs coming to market providing a merging of networks in the home/SOHO environment. The combination of controlling both IoT enabled sensors and home network access from a single interface provides convenience and, in combination with always on, Cloud access, accessibility. The Almond+ ecosystem (including the home hub, iOS/Android companion apps and the Cloud environment) were examined to provide forensic examiners encountering such a device with a method of extraction and analysis. Furthermore, much like any significant changes to technology, the lessons that can be learned from initial adopters can further an analyst's understanding of the broader implications of the technology.

The Almond+ ecosystem has a series of local- and cloud-based logs that are available for analysis. Information is also available from a companion application that provides evidence of connected network devices and user activity.

The Almond+, like the Amazon Alexa service, has the potential to impact on an investigation as the information in the device may have considerable forensic value as it can indicate interaction with the environment at a certain time and date. Observed changes such as humidity, temperature and motion are more passive in their approach. Other changes such as dimming lights, changing room colours and turning wall sockets on and off are more active requiring direct command. The data gathered from an Almond+ system could provide a rich, if textual, picture of events leading up to a crime scene.

During the course of this research several observations were noted that could impact on a user's security and privacy. Default root passwords (which can be viewed in the SecuriFi wiki (Lars, 2014b)) should be changed before the hubs leave the manufacturer as is the case with many router vendors; stickers with a unique code attached to the device for example. The severity would also be lessened by checking the shipping configuration of such devices; as mentioned in Data extraction, the firmware version we assessed had root-accessible SSH access enabled over WiFi by default regardless of what the GUI interface indicated.

Future work should include the analysis of other, similarly marketed home hubs to produce more generalised forensic guide that would work across a range of devices. Furthermore, it would be desirable to develop a forensics tool to analyse the information present on the Almond+ and to correlate the data between the different logs to provide a historical pattern of activity observed by the devices.

Acknowledgements

This research was supported by work funded from the Provost Chase Scholarship Initiatives at Norwich University.

Appendix A. Almond+ file excerpts.
References


