Digital Investigation 28 (2019) S22–S29

Contents lists available at ScienceDirect

Digital Investigation

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

IoT forensic challenges and opportunities for digital traces

Francesco Servida[*], Eoghan Casey

University of Lausanne, 1015, Lausanne, Switzerland

ARTICLE INFO

Article history:

Keywords:
Digital forensics
Internet of things (IoT) forensics
Internet of things (IoT) security
Smartphone forensics
Privacy

ABSTRACT

The increasing number of IoT devices in personal environments such as smarthomes presents opportunities and risks from a forensic perspective. These devices generate traces that can be useful for investigative and forensic purposes in any type of offense. At the same time, newer IoT devices are not supported by existing digital forensic tools and methods, making it difficult for practitioners to extract data from them without the support of a forensic advisor with specialized knowledge in this area. In addition, these traces can present evaluation challenges for forensic scientists, and can contain vulnerabilities that pose privacy risks. Security vulnerabilities of IoT devices create opportunities for extracting traces but might also be used by criminals to undermine a device. The aim of this work is to increase familiarity with traces from various IoT devices in a smarthome, and demonstrate how traces from IoT devices in a smarthome can be useful for investigative and forensic purposes. This work presents a study of IoT devices and associated smartphone applications, providing approaches to extracting and analyzing digital traces. This research led to the discovery of vulnerabilities in multiple devices, and a scenario for the DFRWS IoT forensic challenge was developed.

Introduction

The Internet-of-Things (IoT) is growing rapidly, creating opportunities and challenges for investigators of any type of crime, including cyberattacks and physical assaults (Kebande et al., 2017, Akatyev and James, 2017). By definition and design, smarthomes and other IoT environments are connected, dynamic, and can be altered from anywhere anytime (Minerva et al., 2015; Loung, 2018; Barnard-Wills et al., 2014). Many IoT devices have sensors or actuators that generate data, sometimes autonomously and sometimes in response to human actions (motion detection, door opening). This always active, always generating makes them excellent digital witnesses, capturing traces of activities of potential use in investigations. IoT devices can be invaluable sources of evidence provided digital investigators can manage the quantity of data generated, the number and variety of devices, the heterogeneity of protocols used, and their distributed nature.

Previous research about forensic analysis of IoT devices proposed approaches to facilitate IoT forensics by using frameworks in which traces are proactively collected from the devices and the network to be available for study in case of an IoT related incident (Kebande and Ray, 2016; Zawoad and Hasan, 2015; Amar et al., 2018; Perumal et al., 2015; Dorsemaine et al., 2016). Although such methods are valid, they mainly apply to the industrial IoT sector where the need and the resources exist to put such forensic preparedness in place. Most smarthomes, and even smartbuildings, lack any such forensic preparedness. This work assumes a crime scene that has not been prepared in advance from a forensic perspective. This work concentrates on traces that can be obtained from IoT devices at a crime scene and associated smartphones, proposing a generalized practical process that extends existing methods for examining smartphones.

The devices studied in this work listed in Table 1 were selected on the basis of their popularity in Europe, and specifically Switzerland. Some consideration was also given to the level of potential for extracting traces based on prior research and disclosed vulnerabilities. Traces recovered from the Nest and Wink Hub smartphone applications in this study corresponded to those discovered by Rajewski (2016, 2017). An Amazon Echo and Nest Camera were also included in the scenario development, but were not a focus of study. Prior work already addresses the Amazon ecosystem from a forensic perspective (Chung et al., 2017; Hyde and Moran, 2017).

Prior work that categorized traces from IoT devices included movement, location, temperature, presence/absence, steps taken, distance walk, time spent walking, and calories burnt (Rahman, 2016). During the present study it was possible to extract various kinds of traces from the devices and their associated smartphone

[*] Corresponding author.
E-mail addresses: francesco.servida@unil.ch (F. Servida), eoghan.casey@unil.ch (E. Casey).

https://doi.org/10.1016/j.diin.2019.01.012

Table 1
Devices selected for study in this work.

Manufacturer	Device	Function
Askey	QBee Multi-Sensor Camera	Multifunctional surveillance camera
iSmartAlarm	Cube One & Accessories	Alarm system with a base station, motion sensor, and contact sensor (door)
Netgear	Arlo Pro	Surveillance camera
Nest	Protect	Smoke and CO detector

applications, including system activity logs with details about the events recorded by the device sensors as well as the commands sent by the user. Such traces enable an investigator to infer at what time a door was opened or the moment when the alarm was disabled. Certain devices, such as smoke and carbon monoxide detectors can be useful for determining the time and approximate location where a fire began in a building. Settings of the device and information about the connected sensors or linked devices can be useful to determine the last status of a device. Depending on the devices, this information might or might not survive a reboot. Traces generated by IoT devices are not only present on the physical objects but can also be found on smartphones and the cloud. While traces in the cloud were not part of this research project, traces generated and stored by smartphone applications were.

Digital traces stored in smartphone applications that were used to access and alter the IoT devices in this study included cached image thumbnails and fragments of the camera streams, cached events triggered by the sensors, and complete event logs stored in the application database. These traces provide investigators with information about what happened, when, and which user account sent commands to a device. The image thumbnails could potentially reveal the number of people within the IoT environment, and their identities. Photographs and videos recorded by IoT devices can be highly valuable from a forensic perspective, providing opportunities to attribute physical activities to a specific individual. Cloud credentials were also recovered from smartphone applications, which could be utilized by an investigator to obtain data stored on cloud systems. Importantly, well-established methods for extracting and examining traces from smartphones can be extended to IoT devices, including chip-off and application analysis.

The novel contributions of this research include:

- Extending existing methods for extracting and examining traces from smartphones to IoT devices
- Developing multiple plugins for the Autopsy analysis platform, automating the extraction and parsing of traces related to IoT devices).
- Obtaining memory or filesystem images for the iSmartAlarm and Arlo base stations, as well as the Wink Hub by interrupting the boot process.
- Discovering four previously unknown vulnerabilities on two devices and reporting them to the respective vendors.
- Revealing cloud credentials for two applications, decrypted from the application settings.
- Creating a tool to retrieve and parse log data from the iSmartAlarm base station.
- Conceiving of a method and supporting tool to intercept traffic between the QBee Camera and the smartphone application and reuse it to disable the camera.
- Generating a scenario for the 2018–2019 DFRWS IoT forensic challenge.

Methodology

The study of the devices in this research followed a methodology in six steps: preliminary analysis, testbed setup, network analysis, smartphone application analysis, vulnerability analysis, physical analysis. The objective of this methodology is to allow an investigator to study a new device and discover which traces are available on the device, where and how to collect them, and eventually develop tools to automate the process. This approach extends existing methods for forensic processing of mobile devices (Ayers et al., 2014). The main difference is that the presented methodology deals with a wider variety of device types, in both hardware and software, requiring investigators to adapt forensic principles to deal with new systems and extend existing tools to handle new types of traces.

Preliminary analysis. When an investigator encounters a new IoT device, the first step is to survey existing research on the device, including academic research, security sources such as vulnerability databases, and user community sources. Such a survey can provide information about traces and vulnerabilities that are already known to be on the device, as well as possible ways to gain root access to the device.

Testbed setup. In order to study an IoT device in an extensive way it is important to test it within a controlled environment. This test environment requires a network configuration that enables passive collection of all traffic sent and received by the IoT device. In addition, to test possible attacks on the IoT device, the network configuration needs to support man-in-the-middle (MITM) attacks. To satisfy these requirements, the testing environment in this work used a Raspberry Pi 3 configured to provide internet access to both WiFi and Ethernet devices.

Network Analysis. The objective of analysing network traffic in the test environment is to study the communications to and from the IoT device, including which other devices or systems it communicates with, which communication protocols it uses, and whether readable information is transmitted, either in plaintext or encrypted but vulnerable to MITM attacks. Examining the different entrypoints into the device, the listening ports, and the active services will give insights into possible ways to obtain information from the devices, or the availability of remote access services such as ssh or telnet.

Smartphone Application Analysis. IoT devices, especially ones designed for smarthomes, provide mechanisms for the user to monitor and control the device via the Internet or the local network. These user interactions typically involve a smartphone application that can store information about the device, its configuration, and past events. Analysis of IoT traces stored by associated smartphone applications requires manual investigation of the applications, as mobile forensic software does not, generally, include parsers for these particular applications. In-depth, manual analysis of the smartphone application, including reverse engineering, can uncover additional information which can be correlated with corresponding events recorded by the IoT device and commands sent by the user within the test environment. The results of this in-depth analysis can be codified by writing custom plugins within open source digital forensic tools to automatically process the traces. These plugins will be available for reuse when the device is encountered in future investigations.

Vulnerability Analysis. It is important to analyse vulnerabilities of

Table 2
Summary of findings on each IoT device studied in this work.

Device	Network	Smartphone Applications	Physical	Network Access	Vulnerabilities
ARLO	–	Cloud Credentials (token) Linked Devices Cached Thumbnails	WiFi PSK (Memory Image) Settings and Logs (Root Access)	Telnet Console (From internal network)	–
NEST	–	User Informations Linked Devices Events Video Fragments	Logs via USB (Nest Protect)	–	–
QBee	Cleartext Traffic Port Forwarding with UPnP	Cloud Credentials	–	–	Cleartext Traffic Encrypted Password on Android
iSmartalarm	Diagnostic Logs	Cloud Credentials Recorded Events MQTT Topics Info UPnP Discovered Devices	Memory Images	Diagnostic Logs Access	Unauthenticated Log Access Cleartext Password on Android
Wink	SSDP Advertisements	User Informations Linked Devices Events (Long Term Storage)	Filesystem Images	SSH Console	–
Echo	SSDP Discovery	–	–	–	–

the IoT device to understand how a device could be compromised and exploited by a malicious actor in order to perpetrate a crime, as well as to discover possible ways to access a console on the device in order to acquire data. Vulnerability analysis of IoT devices combines all of the information and data from the preceding steps, and inspects the most common vulnerabilities for the discovered entry points (Open Web Application Security Project, 2014, 2018).

Physical Analysis. When investigating a crime, it is usually not possible to retroactively collect network traffic related to a pertinent IoT device. In such cases, the information stored on the IoT device itself is of paramount importance. Therefore, the final step in the analysis methodology for IoT devices is to perform a physical analysis of the hardware. Depending on the IoT device it could be possible to get access via serial connection (UART) and/or JTAG, or it could be necessary to proceed to chip-off techniques to access the device memory.

Cloud. A further step when dealing with IoT devices in an investigation is to obtain related data from associated cloud service providers. Data from IoT devices is often stored in the cloud for easy retrieval by associated smartphone applications. This data might be available from the cloud service provider or using cloud identifiers recovered from the user phone. Legal authorization is often a prerequisite for obtaining such information in the cloud (James and Jang, 2015).

Extracting traces from IoT devices

During the present research it was possible to extract traces from multiple locations: directly from the memory of the IoT devices, from the network and from the smartphone applications. The results are summarized in Table 2.

Network Access. Although most network traffic associated with the IoT devices in this study was encrypted, some devices communicated with the smartphone application in plaintext. Analysis of network traffic between the QBee device and smartphone application revealed details used to authenticate remote commands to the camera. Through further analysis and testing, it was discovered that these authentication details could be used to remotely control and disable the QBee Camera.[1] When such an attack was performed in the testing environment, it left no trace that could be used later to reconstruct that the attack occurred. Examination of traffic from the iSmartAlarm Cube One found that the diagnostic logs sent from the base station to the smartphone application were transmitted in plaintext as shown in Fig. 1. These logs contain details about events triggered by the sensors, commands received by the base station, and requests made to the iSmartAlarm cloud servers since the device was rebooted. Although these logs could be accessed via the smartphone application, they were not stored or cached on the smartphone. Knowing that the diagnostic endpoint was not authenticated,[2] the collection process was integrated in a Python script to obtain the data directly from the device instead of having to intercept the network traffic to the smartphone application.

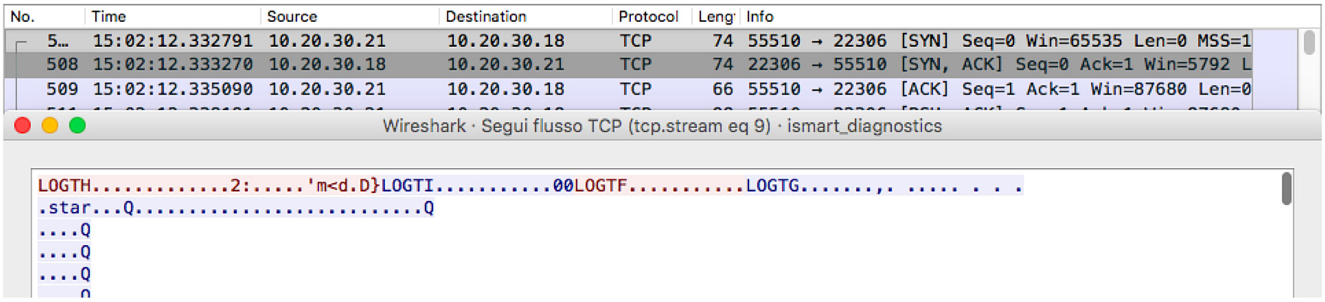
The analysis of the network traffic therefore provided a way to discover how a device could be disabled or otherwise compromised, as well as how recorded event details could be acquired directly from an IoT device.

Direct device access. Utilizing the serial connections on three of the IoT devices studied in this work, it was possible to access the memory/filesystem. Through the bootloader of the iSmartAlarm Cube One and the Netgear Arlo Base Station, it was possible to acquire memory dumps. The memory dump from the Netgear Arlo contained the password for its private WiFi network (The Netgear Arlo creates a custom WPA2 protected WiFi network in order to isolate the cameras from the user's LAN). The iSmartAlarm memory dump did not contain noteworthy traces in the context of this study. By enabling telnet access in the NVRAM settings of the Arlo station and connecting to its private WiFi network, it was possible to obtain root shell access to the device which was used to obtain a partial filesystem copy of the device. The traces collected from the Arlo ranged from details about cameras that were connected to the station and their last settings, to all the logs from the device last factory reset, including for example the timestamps of the motion detection events from the connected camera. The Wink Hub was accessed via the bootloader to obtain a root shell on the Linux system and extract a copy of the full filesystem via SSH. Information in files acquired from the Wink Hub included configuration settings of the station and the connected devices, as well as the system logs. However, in contrast to the Arlo Base station, the system logs were only stored in the RAM and were lost on device reboot. Importantly these traces only concern devices with which the Wink Hub was directly connected, and not devices simply linked to the Wink Hub cloud account.

Smartphone Applications. The ultimate source of traces related to the IoT devices in this study was the smartphone on which the different companion applications were installed. Forensic

[1] CVE-2018-16225.

[2] CVE-2018-16224.


Wireshark · Segui flusso TCP (tcp.stream eq 9) · ismart_diagnostics

```
No.  Time             Source       Destination  Protocol Leng Info
5…   15:02:12.332791  10.20.30.21  10.20.30.18  TCP      74   55510 → 22306 [SYN] Seq=0 Win=65535 Len=0 MSS=1
508  15:02:12.333270  10.20.30.18  10.20.30.21  TCP      74   22306 → 55510 [SYN, ACK] Seq=0 Ack=1 Win=5792 L
509  15:02:12.335090  10.20.30.21  10.20.30.18  TCP      66   55510 → 22306 [ACK] Seq=1 Ack=1 Win=87680 Len=0
```

```
LOGTH.............2:.....'m<d.D}LOGTI..........00LOGTF..........LOGTG.......,. ..... . . .
.star...Q..........................Q
....Q
....Q
....Q
```

Fig. 1. Interception of iSmartAlarm's diagnostic logs.

examination of these smartphone applications revealed multiple types of traces, in configuration files, databases (e.g., SQLite, Realm) and cached files. Details about the events recorded by the sensors were available for the iSmartAlarm application, the Wink Hub and the Nest application. For instance, the iSmartAlarm application recorded event details, commands sent to the device, and which user sent each command. In addition, smartphone applications can save thumbnails and videos extracted from connected camera feeds. For example, such traces were present in the caches of the Arlo and the Nest smartphone applications. Cloud credentials are another significant type of trace stored in some smartphone applications. Smartphone applications for the QBee Camera and the iSmartAlarm stored the username and password in the configuration files, in encrypted and plaintext format, respectively. The smartphone application for the Arlo stored the user ID and an authentication token in a configuration database.

Parsing traces from IoT devices

The traces collected from the IoT devices and smartphone applications posed two main challenges: the quantity of data and the fact that most of it is either unstructured or not explicit (e.g., codes in a database).

During this research multiple parsers were developed,[3] within the open source Autopsy framework as well as standalone ones, to automate the extraction of the traces in a structured and understandable form. Specifically, a plugin was developed to extract cloud credentials from QBee Camera and Swisscom Home App settings on Android. Another plugin was developed to extract cloud credentials, events and user actions from iSmartAlarm settings and database on Android. In addition, a standalone script was created to download the debug logs from the iSmartAlarm base station and parse interesting events.

To develop these plugins, it was first necessary to clean and structure the data when needed. For example, regular expressions were used to parse the interesting logs from the iSmartAlarm diagnostic file. Once the data was structured, it was possible to proceed to the correlation between the known actions performed during testing, and the associated commands and the entries in the data. In this way the different codes in the data were associated with a specific type of action or command. Codifying this knowledge allows for the reuse of the plugins in order to extract automatically from event databases/logs information that could be useful to an investigator.

iSmartAlarm Events. Two tools were developed to parse information related to the iSmartalarm device, a plugin in two parts for Autopsy as well as a standalone python script.

The smartphone application for iSmartAlarm stores data in a SQLite database called iSmartAlarm.DB. The database contains multiple tables, but only two contain pertinent data: TB_IPUDairy and TB_SensorDairy (Table 3). The TB_IPUDairy table (Fig. 2) stores changes in the base station state, user interactions and alarms. The TB_SensorDairy table stores the events generated by the sensors in the latter. These events were mostly in the form of unknown codes. As such, these coded events were compared to the known actions, as documented during the testing in a controlled environment (testbed) and from the interface of the application itself, in order to establish a "translation table" between each code and a specific event/action. The iSmartAlarm application was decompiled with jadx (Skylot, 2013–) and, by determining which functions were responsible for the database management, it was possible to find additional mappings between the codes in the database and the event/action type they represent. Subsequently this data was integrated as a python script in a plugin for Autopsy, which automatically looks for the application database, decodes it, and presents the events in a structured manner.

The standalone script was used to extract and parse the debug logs from the iSmartAlarm device itself. Once obtained, as previously described, the raw logs were a mix of binary data and text data. In the text data (cf. Fig. 3) a common character, "$", was found to be used as a separator between events. Using this knowledge, the python script reads the text data and parses all the unstructured events. Because these events were simple text strings, multiple regular expressions were used to parse the pertinent information and to structure it in a JSON database. The format of the binary part of the file was not deciphered, and is currently omitted during parsing.

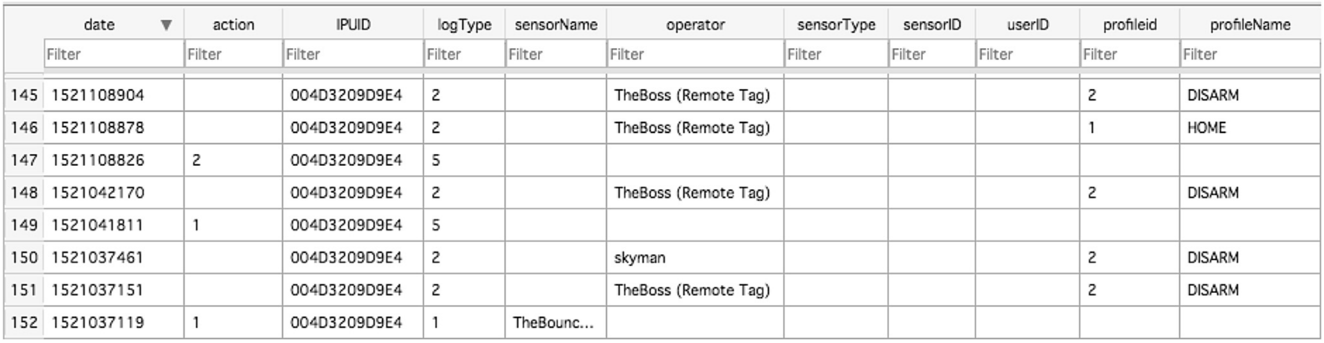
Cloud Credentials. Two other plugins were developed to automate the extraction of user cloud credentials from the applications settings of iSmartalarm and QBee Camera and Swisscom Home App. The iSmartalarm cloud credentials were stored in plaintext in the XML settings file, making them easy to retrieve. Although the QBee Camera could credentials were encrypted, the encrypted settings were in a format suggesting the use of the common "Secure Preferences" library (Alexander-Bown, 2013–). By default, this library stores the AES key alongside the encrypted data in the XML file. However, none of the values was a valid AES key for the decryption. The smartphone application was therefore decompiled to determine which function handles the encryption/decryption of the settings (Fig. 4). The discovered function was a slightly modified version of the "Secure Preferences" one. The AES key is in fact derived from the value in the file, by inserting a hardcoded string and hashing it with SHA 256. With that knowledge it was possible to implement a decryption function in python and integrate it in another Autopsy plugin. Since the same algorithm and hardcoded string were used in the Swisscom Home App, the plugin works for both applications.

[3] https://github.com/fservida/msc_autopsy_plugins.

Table 3
iSmartAlarm - TB_IPUDairy and TB_SensorDairy Translation Tables.

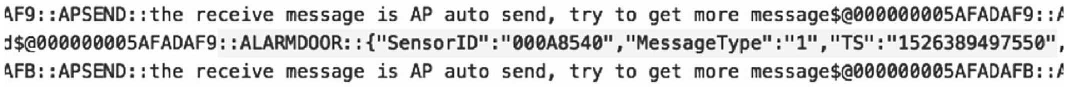
logType	Type	action	Meaning	profileId	profileName
1	Alarm	1	Contact Sensor Alarm	–	–
		2	Motion Sensor Alarm	–	–
2	Profile Change	–	–	0	ARM
		_	–	1	HOME
		_	–	2	DISARM
		_	–	3	PANIC
5	Cube Status	1	Cube Offline	–	–
		2	Cube Online	–	–

logType	Type	action	Meaning
1	RemoteTag Action	1	Arm
		2	Disarm
		3	Back Home
		4	Left Home
_	–	1	Open (Unknown Device)
		2	Closed (Unknown Device)
		3	Door Open
		4	Door Closed
		5	Motion Detection
		6	Low Battery
		7	*Nominal Power*
		8	Sensor Test (model = = 0)
		8	Smoke Detection (model = = 1)
		14	Device Added
		15	Device Deleted



	date	action	IPUID	logType	sensorName	operator	sensorType	sensorID	userID	profileid	profileName
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
145	1521108904		004D3209D9E4	2		TheBoss (Remote Tag)				2	DISARM
146	1521108878		004D3209D9E4	2		TheBoss (Remote Tag)				1	HOME
147	1521108826	2	004D3209D9E4	5							
148	1521042170		004D3209D9E4	2		TheBoss (Remote Tag)				2	DISARM
149	1521041811	1	004D3209D9E4	5							
150	1521037461		004D3209D9E4	2		skyman				2	DISARM
151	1521037151		004D3209D9E4	2		TheBoss (Remote Tag)				2	DISARM
152	1521037119	1	004D3209D9E4	1	TheBounc...						

Fig. 2. iSmartAlarm database (Table IPUDairy).



```
AF9::APSEND::the receive message is AP auto send, try to get more message$@000000005AFADAF9::A
d$@000000005AFADAF9::ALARMDOOR::{"SensorID":"000A8540","MessageType":"1","TS":"1526389497550",
AFB::APSEND::the receive message is AP auto send, try to get more message$@000000005AFADAFB::A
```

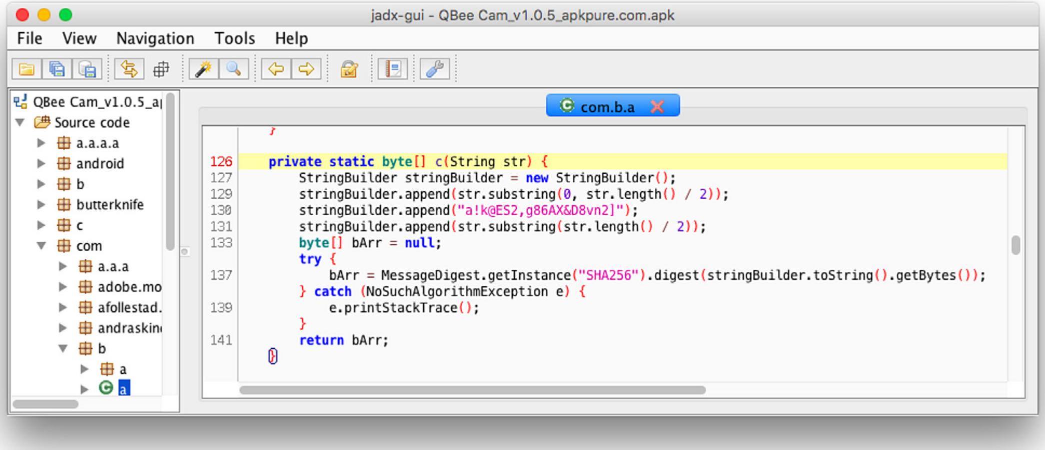
Fig. 3. Door sensor triggered in iSmartAlarm diagnostic logs.

Summary of research findings

This research highlighted two main challenges to the forensic investigation of IoT devices.

The first challenge relates to the analysis of the network traffic. An increasing amount of traffic is encrypted, which is a beneficial development for the security of the users, but limits the collection of interesting traces to those transmitted by the less secure IoT devices. Additionally, IoT devices do not limit communication to the WiFi and Ethernet protocols studied in this research; a number of devices use protocols such as ZigBee, Z-Wave, Bluetooth or custom radio frequencies protocols for the communication between the sensors and the base station. The analysis of these additional protocols could provide new traces or, more importantly, indications of possible ways to exploit vulnerabilities in the devices, but requires more complex equipment and extended expertise.

The second challenge concerns the traces present on the physical devices. While traces were obtained from the devices, these traces either limited themselves to configuration settings or had limited persistence, mainly due to the limited storage available. The traces about the events recorded by the devices for example were retained only until a reboot on the iSmartAlarm Cube One and Wink Hub. The Arlo base station was the most interesting from a forensic point of view because the traces were retained until a factory reset. Moreover, access to these traces is non-trivial at the moment; the technique used in this study exploited the access to the bootloader via serial connection to extract memory images or obtain a root shell in order to extract a filesystem image. Other techniques include using JTAG or chip-off to obtain access to the data on the device. These procedures can be technically challenging, especially for an investigator with limited knowledge of the field, and can be destructive for the device and the traces on it.



```java
    }

    private static byte[] c(String str) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append(str.substring(0, str.length() / 2));
        stringBuilder.append("a!k@ES2,g86AX&D8vn2]");
        stringBuilder.append(str.substring(str.length() / 2));
        byte[] bArr = null;
        try {
            bArr = MessageDigest.getInstance("SHA256").digest(stringBuilder.toString().getBytes());
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return bArr;
    }
```

Fig. 4. Decryption function in QBee Camera application.

It is foreseeable that future developments in forensic techniques and technology could provide simplified ways to perform a physical extraction of data from IoT devices, similar to ongoing advances in smartphone forensics.

Given the current challenges of the physical analysis of IoT devices, smartphone and cloud forensics are complementary. Indeed, most of the data will be sent to the cloud for easy retrieval from the smartphone applications/webpages, and most of that data will subsequently be synced to the mobile device.

The analysis of the linked smartphone(s) can therefore produce traces not available anymore on the device itself. The case of the iSmartAlarm is the most flagrant discovered in this research, after a reboot the logs about the time and date of the events would be lost on the base station; however parsing the application's database on the smartphone would yield the totality of events cached from the cloud during a normal usage of the application, which could extend up to the moment the device was setup. The main problem with forensic analysis of the traces on a smartphone is that some information, particularly for recent activities, may not be available because they have not been automatically synced with the associated applications.

This problem can partially be solved by cloud forensics. By accessing the data in the cloud, either with a legal request to the cloud service provider or authorized use of credentials extracted from the smartphone, an investigator could download the available data, including the most recent events, not yet synced with the smartphone. This approach can also allow, for example, the download of more data than could be retrieved from a device. For instance, in the case of the Arlo, it was possible to retrieve the full recorded clips, instead of merely the thumbnails available on the smartphone. One downside of this approach is that some services, for example the Nest and the Arlo, offer subscriptions to the user, and the data on the cloud is retained only during a certain timeframe depending on the subscription plan (the exception being eventual backup copies still available). In this context, aggregation devices and services such as those offered by the Wink Hub are of extreme interest from a forensic point of view; these services poll the linked devices or accounts and retain a copy of the event details in separate cloud system. This data can remain on the separate cloud system even after it has been removed from the original cloud system, either because it exceeded the allowed timespan or because it could have been intentionally deleted by the user.

Changeability & accessibility

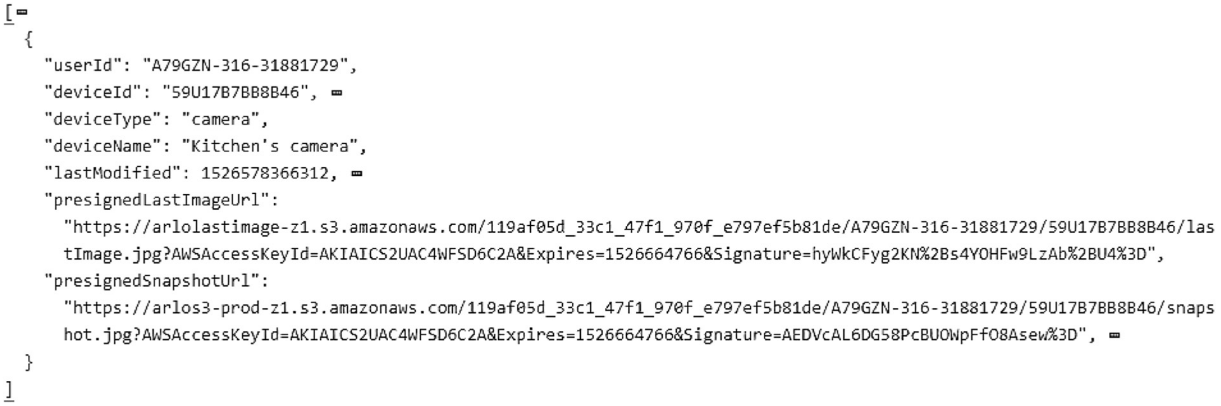
The same features that make IoT devices excellent digital witnesses, can create challenges for forensic preservation - the moment digital investigators approach a space monitored by IoT devices, they are generating traces, virtually stepping in the digital evidence. Such alterations are a form of evidence dynamics (Casey, 2011). Evidence dynamics can occur in any crime scene, impacting physical and digital evidence, but is particularly pronounced in IoT equipped environments. For instance, some devices and related applications store information about the last event, or a token to access such information or images on the cloud. By interacting with the scene and the devices, the last event will no longer be related to the events of interest but instead to the events generated by the investigators as shown in Fig. 5 for the Arlo camera.

Another point to consider is that as the devices have low storage capabilities it is possible that the events generated by the investigators/first responders fill the device memory, prompting it to prune the older, relevant events. The additional data created by the intervention on the scene means also that the traces will be polluted with non-relevant data; this underscores the need for extensive documentation of all interaction with the devices on the scene by first responders and investigators alike, to be able afterward to differentiate between irrelevant and pertinent traces.

These issues are challenging not only for investigators in a real case, but also when creating a forensic challenge scenario. In fact, it is difficult in such an environment not to leave the touch of the creator in the traces. The preparation of the scenario for the 2018–2019 DFRWS IoT challenge provided some examples of this problem.

A first difficulty consisted in the unwanted triggering of a motion detection; shortly after the scenario was concluded, an erroneous passage in front of the Arlo camera meant the device, and the smartphone application, showed traces of this passage, which was not planned in the scenario. As the data from the Arlo device was not yet obtained, nor the smartphone already imaged, it meant that the scenario had to be modified to account for the presence of somebody after the crime happened but before the police arrived on scene; in that case by the addition of a witness who entered the laboratory.

Another problem derives from the smartphone application of iSmartAlarm. That application sent PNP discovery requests on the network to discover nearby devices, which resulted in traces in the



```
[
  {
    "userId": "A79GZN-316-31881729",
    "deviceId": "59U17B7BB8B46",
    "deviceType": "camera",
    "deviceName": "Kitchen's camera",
    "lastModified": 1526578366312,
    "presignedLastImageUrl":
      "https://arlolastimage-z1.s3.amazonaws.com/119af05d_33c1_47f1_970f_e797ef5b81de/A79GZN-316-31881729/59U17B7BB8B46/las
      tImage.jpg?AWSAccessKeyId=AKIAICS2UAC4WFSD6C2A&Expires=1526664766&Signature=hyWkCFyg2KN%2Bs4YOHFw9LzAb%2BU4%3D",
    "presignedSnapshotUrl":
      "https://arlos3-prod-z1.s3.amazonaws.com/119af05d_33c1_47f1_970f_e797ef5b81de/A79GZN-316-31881729/59U17B7BB8B46/snaps
      hot.jpg?AWSAccessKeyId=AKIAICS2UAC4WFSD6C2A&Expires=1526664766&Signature=AEDVcAL6DG58PcBUOWpFfO8Asew%3D",
  }
]
```

Fig. 5. Traces of the last photograph taken by an Arlo surveillance camera now relate to the investigators present at the crime scene, not the previous photograph that existed prior to their arrival at the scene.

configuration files suggesting the presence of a specific network printer on scene. The printer however was not on scene but on a different network used to test remote access to the devices. While this could be useful information in an investigation for locations frequented by the smartphone user, it was unwarranted in the scenario.

Lastly, the configuration of some devices, notably the NEST camera, could not be performed directly on an iOS device (an iPhone SE, initially used for the study but shortly replaced by the Samsung phone), and was therefore made using a computer, with a dedicated Chrome profile. The separation on the profile was not however sufficient, by error some personal pages were opened in that profile, and after imaging the phone for the scenario it became evident that personal information and history were synced to the chrome browser on the phone (both used the same Google account: jessie.pinkman@gmail.com).

Accessibility of the traces on the devices is also problematic, as outlined above, depending on the device it is possible to obtain much more information if the device is examined prior to power-off. However, this requires preexisting knowledge of the specific IoT device and any applicable tools for processing traces. For an investigator arriving on a scene this means that three main situations present themselves: she/he knows the device and already has some tools to extract directly most of the data, she/he knows that there is no way to rapidly extract the live data or she/he does not know the device at all. In the first situation the investigator will be able to utilize said tools on scene, and transport the device back to the forensic laboratory for preservation and additional analysis. In the last two situations the investigator will most likely have to bring the device back to the laboratory for advanced analysis, losing data in volatile storage with the risk that no copy was available on the persistent storage. This situation highlights the need for pro-active study of devices that could frequently be encountered on a scene; since most of the interesting data (event logs, user activities) are stored on volatile memory and digital investigators want to maximize the chances of recovering these traces.

Attribution and evaluation

A final challenge with IoT traces is establishing a link between the digital traces and physical activities and entities. IoT traces can be used to reconstruct activities in great detail, but an incorrect assumption or overlooked event can lead to the wrong conclusion.

For example, consider a violent crime in a smarthome as was simulated in the DFRWS 2017–2018 IoT Forensic Challenge.[4] One of the questions “Evaluating and Expressing Conclusions: Assigning the probability of the results given two competing propositions (e.g. The husband killed the wife, some unknown person did).” Although traces on IoT devices in the victim's home might indicate that the victim fought with a man other than her husband immediately before her death, this does not prove that the unknown man murdered the victim. A common mistake is to concentrate on a specific hypothesis, such as the husband killed his wife. Such strength of hypothesis approach is to be avoided (Casey, 2018). Alternative possibilities must be considered carefully, including the victim harmed herself, the husband killed his wife, or the victim is still alive. The proper approach is to evaluate the strength of evidence given at least two propositions.

As another example, when someone speaks commands to an Amazon Echo device, additional analysis is necessary to determine whose voice it was, and whether the person was physically close to the device or speaking through audio conference from a different location.

Presence indicating events such as motion detection or door opening are not directly linked to an identity. When a trace from an IoT device shows that a specific user deactivated a security alarm before the front door was opened and someone entered, this could provide a strong indication that the account owner issued the command. However, the account owner could have issued the command remotely to allow someone else to enter the front door.

Obtaining the smartphone that interacted with IoT devices can provide a more a complete picture of an individual's presence in a particular place at a specific time. However, even events within smartphone applications do not necessarily mean the presence of a user. When geolocation information indicates that the device was at the scene when the crime occurred, additional analysis is needed to determine if the clock was correct, the location information was accurate, and that the device was not being used by someone else at the time. When IoT devices generate multimedia traces such as photographs or audio recordings, face or voice comparison can provide stronger evidence of identity and attribution of activities. In any investigation involving IoT devices, care must be taken to evaluate the digital traces under multiple alternative explanations.

Forensic considerations

To handle IoT devices properly from a forensic perspective, investigators could be assisted by forensic advisors with specialized digital forensic expertise, helping them recognize, preserve and

[4] https://github.com/dfrws/dfrws2017-challenge.

prioritize digital traces at the crime scene. The effectiveness of forensic advisors has been demonstrated in disciplines other than digital forensics, providing guidance and expertise throughout the investigative process, including which sources of evidence and forensic processes could be most valuable for their case, and what questions to ask specialists in the forensic science laboratory (Bitzer et al., 2018). Ideally digital/multimedia forensic advisors should have broad expertise in the different fields of forensics, not limiting themselves to digital disciplines; with a transversal education (Burri et al., 2018) they will be able to support the investigation and know when to coordinate with technical specialists for in depth analysis.

This work also helps address questions of admissibility of evidence from IoT devices in court. The research results for the specific devices studied in this work will help with the evaluation and interpretation of future evidence presented in court about the same devices, by providing a reference for otherwise unknown devices. More generally, when a new IoT device contains traces that will be presented as evidence in court, it is beneficial to study the device following the methodology provided in this work in order to understand the meaning and limitations of the traces. Utilizing this methodology to research other popular devices will extend digital forensics capabilities to evaluate evidence from the broader set of IoT devices properly.

Conclusion and future work

The rising prevalence of IoT devices in homes and buildings increases the opportunities to recover digital traces that are relevant to an investigation, whether it be physical (e.g., burglary, arson) or virtual (e.g., cyberattack, identity theft). This work demonstrates that, in addition to traces on smartphone applications, there can also be useful traces stored on IoT devices themselves. Mobile device forensic methods can be applied in general to extract and examine traces from IoT devices, but the variety of IoT platforms sometimes require device-specific approaches. As demonstrated in this work, open source forensic tools can be customized to process traces from IoT devices. For digital forensics to rekeep pace with technological developments, there is a pressing need for more research into IoT devices and their associated smartphone applications, the traces they generate and contain, and security vulnerabilities that open privacy concerns as well as forensic extraction opportunities. In particular, there is a need for more in-depth physical analysis of IoT devices, including chip-off techniques commonly applied to mobile devices. Additional study is needed of the most common home security systems, smart assistants as well as smart firewalls.

The potential criminal exploitation of IoT devices must also be considered. A criminal can use information generated by IoT devices to stalk a victim or plan an attack, such as accessing surveillance cameras and other IoT systems in a home when planning a robbery (determining when the owners are not in their home), and misusing or disabling IoT devices to prevent them from recording events related to a crime.

Acknowledgements

We would like to thank DFRWS for the support of this study and of the development of the IoT Forensic Challenge. We would also like to thank Seculabs SA [5] for collaborating on the physical analysis and vulnerability assessment of the devices.

[5] https://www.seculabs.ch.

References

Akatyev, Nikolay, James, Joshua I., 2017. Evidence identification in iot networks based on threat assessment. Fut. Gen. Computer Syst. https://doi.org/10.1016/j:future:2017:10:012. ISSN 0167–739X. http://www:sciencedirect:com/science/article/pii/S0167739X17300857.

Alexander-Bown, Scott, 2013. Secure-preferences. https://github.com/scottyab/secure-preferences.

Amar, Yousef, Haddadi, Hamed, Mortier, Richard, Brown, Anthony, James, A., Colley, Crabtree, Andy, 2018. An analysis of home iot network traffic and behaviour. CoRR abs/1803.05368. http://arxiv.org/abs/1803.05368.

Ayers, Rick, may 2014. Sam Brothers, and Wayne Jansen. Guidelines on Mobile Device Forensics. Technical Report. NIST Special Publication, 800-101 Rev 1.

Barnard-Wills, David, Marinos, Louis, Portesi, Silvia, 2014. Threat Landscape and Good Practice Guide for Smart Home and Converged Media. Technical report. European Union Agency for Network and Information Security (ENISA). https://www:enisa:europa:eu/publications/threat-landscape-for-smart-home-and-media-convergence.

Bitzer, Sonja, Heudt, Laetitia, Barret, Aurélie, George, Lore, Karolien Van Dijk, Gason, Fabrice, Renard, Bertrand, 2018. The introduction of forensic advisors in Belgium and their role in the criminal justice system. Sci. Justice. ISSN: 1355-0306 58 (3), 177–184. https://doi.org/10.1016/j.scijus.2017.11.002. http://www.sciencedirect.com/science/article/pii/S135503061730120X.

Burri, Xavier, Servida, Francesco, Vincart, Adrien, David-Olivier, Jaquet-Chiffelle, Casey, Eoghan, 2018. A New Forensic Orientation at Esc to Handle Digital Traces. https://doi.org/10.13140/rg.2.2.18386.45768.

Casey, Eoghan, 2011. Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet, third ed. Academic Press, Inc., Orlando, FL, USA, ISBN 0123742684. 9780123742681.

Casey, Eoghan, mar 2018. Clearly conveying digital forensic results. Digit. Invest. 24 (1–3) https://doi.org/10.1016/j.diin.2018.03.001.

Chung, Hyunji, Park, Jungheum, Lee, Sangjin, 2017. Digital forensic approaches for amazon alexa ecosystem. Digit. Invest. ISSN: 1742-2876 22, S15–S25. https://doi.org/10.1016/j.diin.2017.06.010. http://www.sciencedirect.com/science/article/pii/S1742287617301974.

Dorsemaine, Bruno, Gaulier, Jean-Philippe, Wary, Jean-Philippe, Kheir, Nizar, Urien, Pascal, 2016. A new approach to investigate iot threats based on a four layer model. In: 13th Int. Conf. New Technol. Distrib. Syst. IEEE, pp. 1–6. https://doi.org/10.1109/NOTERE:2016:7745830. ISBN 978-1-5090-3426-0. http://ieeexplore:ieee:org/document/7745830/.

Hyde, Jessica, Moran, Brian, 2017. Alexa, are you skynet?. In: SANS DFIR Summit 2017 https://www:sans:org/summit-archives/file/summit-archive-1498230402:pdf.

James, Joshua I., Jang, Yunsik, 2015. Practical and legal challenges of cloud investigations. J. Inst. Internet, Broadcast. Commun. https://doi.org/10.7236/JIIBC.2014.14.6.33.

Kebande, Victor R., Ray, Indrakshi, aug 2016. A generic digital forensic investigation framework for internet of things (iot). In: 2016 IEEE 4th Int. Conf. Futur. Internet Things Cloud. IEEE, ISBN 978-1-5090-4052-0, pp. 356–362. https://doi.org/10.1109/FiCloud.2016.57. http://ieeexplore.ieee.org/document/7575885/.

Kebande, Victor R., Karie, Nickson M., Michael, Antonia, Malapane, Semaka M.G., Venter, H.S., 2017. How an iot-enabled \smart re-frigerator” can play a clandestine role in perpetuating cyber-crime. In: IST-Africa Week Conf. IEEE, pp. 1–10. https://doi.org/10.23919/ISTAFRICA:2017:8102362. http://ieeexplore:ieee:org/document/8102362/.

Loung, Tony, 2018. Thermostats, locks and lights: digital tools of domestic abuse. New York Times 6. https://www:nytimes:com/2018/06/23/technology/smart-home-devices-domestic-abuse:html.

Minerva, Roberto, Biru, Abyi, Rotondi, Domenico, 2015. Define Iot - Ieee Internet of Things. Technical Report. https://iot.ieee.org/definition.html.

Open Web Application Security Project, 2014. Top iot vulnerabilities. https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.

Open Web Application Security Project, 2018. Iot attack surface areas project. https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Attack_Surface_Areas.

Perumal, Sundresan, Norwawi, Norita Md, Raman, Valliappan, 2015. Internet of things(iot) digital forensic investigation model: top-down forensic approach methodology. In: Fifth Int. Conf. Digit. Inf. Process. Commun. IEEE, pp. 19–23. https://doi.org/10.1109/ICDIPC:2015:7323000. ISBN. 978-1-4673-6832-2. http://ieeexplore:ieee:org/document/7323000/.

Rahman, Sabidur, 09 2016. Internet of things mobility forensics. In: Proceedings of the 2016 Information Security Research and Education (INSuRE) Conference (INSuRECon-16).

Rajewski, Jonathan T., 2016. Internet of things forensics. In: Enfuse 2016. https://www.jonrajewski.com/data/Presentations/EnFuse2016/Enfuse_2016_Internet_ofΩThings_Rajewski.pdf.

Rajewski, Jonathan T., 2017. Internet of things forensics. In: Enfuse 2017. https://www.jonrajewski.com/wp-content/uploads//2017/07/Enfuse_2017_Rajewski_IOT_Forensics.pdf.

Skylot Jadx, 2013–. URL https://github.com/skylot/jadx.

Zawoad, Shams, Hasan, Ragib, jun 2015. Faiot: towards building a forensics aware eco system for the internet of things. In: 2015 IEEE Int. Conf. Serv. Comput. IEEE, ISBN 978-1-4673-7281-7, pp. 279–284. https://doi.org/10.1109/SCC.2015.46. http://ieeexplore.ieee.org/document/7207364/.