Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

Knock! Knock! Who is There? Investigating Data Leakage from a Medical Internet of Things Hijacking Attack

Talon Flynn
School of Interdisciplinary Informatics
University of Nebraska at Omaha
tflynn@unomaha.edu

George Grispos
School of Interdisciplinary Informatics
University of Nebraska at Omaha
ggrispos@unomaha.edu

William Bradley Glisson
Digital Forensics and Cybersecurity Research Center
Sam Houston State University
glisson@shsu.edu

William Mahoney
School of Interdisciplinary Informatics
University of Nebraska at Omaha
wmahoney@unomaha.edu

Abstract

The amalgamation of Medical Internet of Things (MIoT) devices into everyday life is influencing the landscape of modern medicine. The implementation of these devices potentially alleviates the pressures and physical demands of healthcare systems through the remote monitoring of patients. However, there are concerns that the emergence of MIoT ecosystems is introducing an assortment of security and privacy challenges. While previous research has shown that multiple vulnerabilities exist within MIoT devices, minimal research investigates potential data leakage from MIoT devices through hijacking attacks.

The research contribution of this paper is twofold. First, it provides a proof of concept that certain MIoT devices and their accompanying smartphone applications are vulnerable to hijacking attacks. Second, it highlights the effectiveness of using digital forensics tools as a lens to identify patient and medical device information on a hijacker's smartphone.

1. Introduction

The integration of wireless communication capabilities is dramatically influencing the landscape of modern medicine. This evolution is introducing medical devices that will operate more efficiently, safely, and securely over wireless networks [1]. Recent studies by the European Commission and IBM estimate that within the next decade, over 50 billion medical devices will be Internet capable [2, 3]. Coupling this information with industry predictions indicating that 49% of individuals own a wearable device, supports the idea that individuals are increasingly interested in monitoring their health, medical, and dietary practices [4, 5].

While Medical Internet of Things (MIoT) devices are often for personal use, they are also useful in larger medical environments [6]. Within hospitals, these devices can record and collect patient data and integrate these measurements into Electronic Health Records (EHRs). Hence, both individual and hospital MIoT devices potentially produce and collect vast amounts of medical and patient information. For example, an Internet-enabled next-generation ventilator is expected to generate almost 305 data parameters per second [7, 8]. As a result of these predictions, a Stanford Medicine report goes on to estimate that the medical industry will generate 2,314 exabytes of data by 2020 [9].

However, this data explosion in medical environments also introduces an assortment of security and privacy challenges, from both industry and academic perspectives. Patient information, therapy details, and device operation metadata generated and collected by MIoT devices are all considered to be private information according to the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [10]. As a result, the Security and Privacy Rules within HIPAA specify that entities "maintain appropriate administrative, technical, and physical safeguards for protecting electronic Patient Health Information (ePHI)" [11, 12]. HIPAA additionally states that entities are required to "preserve the confidentiality, integrity, and availability of collected ePHI data, as well as protecting against malicious users and unauthorized disclosures" [10].

URI: https://hdl.handle.net/10125/64533
978-0-9981331-3-3

Many MIoT devices also interface to a smartphone application. This application allows users to track personal and medical information communications with a MIoT device. However, previous research demonstrates that smartphones and their accompanying applications can contain user-related residual data [13-17]. From the perspective of a medical device, research indicates that smartphone residual data can be used to identify metadata related to a specific patient and their interactions with the medical device itself [18]. Further complicating matters, researchers have established that residual artifacts generated by smartphone applications can be used to identify broad user behavior patterns [19].

Current research also indicates that medical devices are susceptible to cleartext network transmissions, often without leaving a trace [20, 21]. The marriage of smartphone applications with MIoT devices, coupled with both the growing volume of data and the identified security and privacy concerns prompts the idea that these devices are vulnerable to hijacking attacks. This idea prompted the hypothesis that *MIoT devices are susceptive to hijacking attacks, through their accompanying smartphone applications*. This hypothesis also raises the following supplementary research questions:

- Does a hijacking attack generate recoverable residual data?
- If so, is it possible to recover MIoT device readings from a hijacking attack involving the device?
- If residual data does exist, is it possible for an attacker to identify a specific individual using information from a hijacking attack?

The contribution of this research is twofold. First, it provides a proof of concept that certain MIoT devices and their accompanying smartphone applications are vulnerable to hijacking attacks. Second, it highlights the effectiveness of using digital forensics tools as a lens to identify patient and medical device information on a hijacker's smartphone. The balance of the paper is structured as follows. Section 2 presents related work, and Section 3 presents the methodology employed in this research. Section 4 presents the results and a discussion of these results. Section 5 derives conclusions and presents ideas for future research.

Related Work

The continuous integration of technology in medical settings is creating an environment where medical devices are potentially at risk from a security perspective [22]. Complicating matters, research indicates that residual data from mobile and GPS devices are used in civil and criminal legal contexts and that there are legal issues around conducting cloud investigations [23-25]. The potential critical impact on human life, coupled with legal implications, encourages discussions by researchers on the security implications of technology in hospital environments [26-28]. Malasri and Wang [26] argue that implantable medical devices, such as pacemakers, are susceptible to a variety of attacks, including eavesdropping, patient tracking, and spoofing. For example, an individual could send malicious commands to compromise the security of a pacemaker, causing direct physical harm to an individual [26]. Glisson et al. [27] demonstrate how a medical mannequin, within a hospital environment, could be vulnerable to denial of service and brute force attacks. Li et al. [28] focus their research on diabetes therapy devices. These researchers argue that some devices transmit patient and device information in plaintext, including passwords and dosage information.

The emergence of Medical Internet of Things (MIoT) ecosystems is expected to introduce several benefits and opportunities for the medical and healthcare communities [29-33]. MIoT devices are defined as "a group of devices connected to the Internet, to perform the processes and services required to support healthcare" [34]. Baker et al. [29] claim that MIoT devices provide a potential solution that alleviates pressures and physical demands on healthcare systems through the remote monitoring of patients. For instance, MIoT devices could monitor patients in remote and rural areas, as well as elderly patients, from the comfort of their home [29].

Dimitrov [31] contends that deploying MIoT devices to patients allows medical practitioners to provide personalized and customized treatment plans. Separately, medical researchers contend that providing anxious patients with MIoT devices, for home use, could provide more accurate and reliable medical results [30, 32, 33].

While the benefits of deploying MIoT devices are clear, there are concerns that patient and medical information could be vulnerable to attack by malicious users. This concern is particularly true when MIoT devices are used in environments where it is difficult to control the underlying network, such as a public Wi-Fi hotspot [34]. Williams and McCauley [35] add that because MIoT devices collect large amounts of personal and health information, these devices are more likely to be targeted by malicious users and cybercriminals. The collection of vast amounts of data introduces the threat of cross-linking information and subsequently using this information to draw conclusions about a patient [35]. Hence, several researchers have focused their efforts on examining the security and privacy challenges that emerge from the deployment of MIoT devices in both hospital and home settings [20, 21, 36-39].

Lotfy and Hale [38] studied the data exchange mechanisms used within various health wearable devices, from a security perspective. The focus of the study was to investigate the Bluetooth Low Energy (BLE) pairing strategies in three devices, a Jawbone, a Pebble Watch, and a Fitbit. Their results show that while manufacturers claim that their pairing strategies are secure, vulnerabilities exist that could result in man-in-the-middle attacks.

Fereidooni et al. [37] focused their efforts on the security of seventeen fitness tracking products. Their attack focuses on the data exchanged between the fitness tracker's smartphone applications and the manufacturer's cloud service. These researchers successfully demonstrated how a malicious user could inject fabricated data into spoofed medical activity records [37].

Alisgari et al. [36] examined security weaknesses in mobile health smartphone applications, which are often used together with MIoT devices. This analysis investigated the use of the Transport Layer Security protocol in twenty-five mobile health applications. Alisgari et al. [36] reported that twenty-one out of the twenty-five applications were susceptible to man-in-the-middle attacks. Moreover, the results of the analysis revealed that twelve applications leaked the user password during network transmissions.

Wood et al. [20] investigated an attacker's ability to intercept MIoT data transmissions, and to then use this information to build a profile of the user. This analysis focused on analyzing network packets transmitted by four MIoT devices. The results showed that information captured from one of the devices included sensitive user information, which would allow an attacker to determine not just that the user measured their blood pressure, but also how frequently the user was taking these measurements. While the packet analysis does not identify individual names, a unique user identifier was recognized during the analysis of the packet transmission [20].

Classen et al. [21] analyzed the entire Fitbit eco-system, including its smartphone application, the Fitbit cloud, and the Fitbit's device firmware. Through their analysis of these technologies, Classen et al. [21] explained that multiple vulnerabilities exist, which could impact a user's privacy and the security of their information. To mitigate these concerns, Classen et al. [21] suggested that Fitbit implement security by design principles and stronger encryption on the smartphone application.

Siddiqi et al. [39] focused on timestamps and their vulnerability to modification in MIoT devices. The authors demonstrated how an attacker could, potentially, intercept and modify medical and patient information, before it is stored in the cloud. This includes timestamp information, which would allow an attacker to backfill medical data and commit insurance fraud [39]. While previous research has examined a variety of security vulnerabilities in MIoT devices, minimal research investigates the ability for an individual to undertake a hijacking attack using a smartphone application and its corresponding MIoT device.

3. Experiment Design

To investigate the hypothesis and associated research questions identified in the introduction, a controlled experiment was conducted as described by Oates [40]. The controlled experiment consisted of eight stages. The eight stages included: 1) preparing the victim smartphones, installing the Medical Internet of Things (MIoT) device smartphone applications**,** as well as creating test accounts for use in the experiment; 2) synchronizing the MIoT devices with the victim smartphones and then powering down these smartphones; 3) using the MIoT devices; 4) preparing a hijacker smartphone device, installing the MIoT device smartphone applications and setting up a test account; 5) executing the MIoT smartphone applications on the hijacker smartphone and attempting to obtain offline readings from the MIoT devices; 6) conducting a manual examination of the hijacker smartphone; 7) processing the hijacker smartphone using MicroSystemation (MSAB) XRY to create an extraction dump; 8) using forensic tools to extract files and artifacts from the extraction dump.

The smartphones utilized in this experiment include a Samsung Galaxy S6 and an Apple iPhone SE (hereafter referred to as the victim smartphones) and a Samsung Galaxy S4 (hereafter referred to as the hijacker smartphone). Table 1 - Smartphone Devices presents an overview of these devices, their features, and storage capabilities.

Feature	Galaxy S4	Galaxy S6	iPhone SE
Model Number	SGH-i337	SM-G920P	A1662
Operating System	Android v. 5.0	Android v. 7.0	iOS v. 11.4.1
Storage Capacity	16 GB	32 GB	32 GB

Table 1: Smartphone Devices

The victim smartphones were selected based on the operating systems executed on the devices. The Android and iOS operating systems represent the two most popular smartphone operating systems at the time of the research [41]. The hijacker smartphone was selected based on its compatibility with the XRY forensic toolkit, which was used to extract a memory dump of the device's internal memory. Several smartphones could have been used to fulfill these criteria and could have been used in the research. The decision to use these specific devices was based on author availability.

The Medical Internet of Things (MIoT) devices used in this experiment includes an iHealth Smart glucometer, an iHealth Air oximeter, and a Nokia Body scale. Table 2 - MIoT Devices, presents an overview of these devices, their model numbers, and firmware versions. These devices were selected based on two reasons. First, all three MIoT devices include both an Android and iOS smartphone application, which can be executed on the victim and hijacker smartphones. Second, each MIoT device can store offline readings, when the user's smartphone application is not available to 'push' the results to the smartphone interface. The MIoT devices include a specific smartphone application. For the glucometer, the application used was Gluco-Smart (v. 4.7 on both Android and iOS), for the oximeter, the application used was MyVitals (Android v. 3.8.1 and iOS v. 3.8), and for the scale, the application used was Health Mate (Android v. 3.5.4 and iOS v. 4.0.1).

Device Name	Model Number	Firmware
Smart Glucometer	BG5	V. 6.0.0
Air Oximeter	PO3M	V. 2.1.4
Body Scale	03700546702341	V. 1751

Table 2: MIoT Devices

In preparation for the experiment, the victim and hijacker smartphones were 'hard reset' to remove any previous data. The purpose of the hard reset is to restore the factory settings on smartphones. Depending on the smartphone, either a Google or Apple account was then created on the smartphone to complete the initial setup. All default setup options were selected during this process. The following steps were then undertaken during the experiment, which involved the victim smartphones, the MIoT devices, and the hijacker smartphone:

1. The victim smartphones were connected to a local wireless network for the experiment. This wireless network was used to access the Internet. Using the victim smartphone's respective application store (i.e., Google Play and the Apple App Store), the MIoT smartphone applications were downloaded and installed on each of the victim smartphones. The default installation and security parameters were used to install the smartphone applications.
2. The MIoT smartphone applications were executed on each victim smartphone, and test profile accounts were created for the experiment. These profile accounts used test information to complete a user profile, which included: first name, last name, date of birth, gender, and email address fields. Default settings were used to complete the profile creation on all three MIoT applications.
3. After setting up each MIoT smartphone application, the user interface was used to 'pair' the victim smartphone with the corresponding MIoT device. This involved using the respective smartphone application to 'search' for the corresponding MIoT device. After the device was found, the application interface provided steps to confirm the 'pairing' of the MIoT devices with the victim smartphones. At this point, the smartphone applications were 'ready' to receive device readings from the MIoT devices. Both victim smartphones were then powered down.
4. Each MIoT device was then used once a day for six days. The first three days involved undertaking readings using the device's iOS application profile. The last three days involved undertaking readings using the device's Android application profile. The medical information measured using each MIoT device was as follows:
   - Glucometer – blood sugar level
   - Oximeter – oxygen level and pulse
   - Scale – weight, body fat percentage, water percentage, pulse, bone weight, muscle weight, and the Body Mass Index value.

   The device reading, as displayed on the MIoT device interface, along with the date and time of each reading, was documented for later analysis.
5. The Android hijacker smartphone was then connected to the wireless network to gain access to the Internet. The relevant MIoT applications were then installed on the hijacker's smartphone, using the default installation and security parameters to complete the installation. The MIoT smartphone applications were executed on the hijacker smartphone, and test profile accounts were created for the experiment. After the profiles were created, the hijacker smartphone applications were used to 'search' for the MIoT devices. When a MIoT device was found, an attempt was made to 'pair' the hijacker smartphone with the MIoT device. If the pairing was successful, the hijacker's smartphone application was then prompted to 'download' any available offline readings. This process was repeated for all three MIoT applications.

6. Immediately after the hijacker smartphone attempted to ‘download’ the offline readings, the smartphone contents were scrutinized using a manual mobile phone forensics examination technique [42]. This involved examining the MIoT smartphone application interface to determine if the victim’s medical information was visible through the smartphone interface. The hijacker smartphone was then processed using MSAB’s XRY (version 7.7) mobile forensic toolkit. The XRY toolkit was used to create a forensic extraction of the smartphone’s internal memory. A wizard provided instructions on how to prepare the device for the extraction. The hijacker’s smartphone internal memory was then read, and a memory dump was saved to a desktop folder on the forensic workstation. The overall process took approximately thirty-five minutes.
7. The forensic extractions were loaded into XRY’s associated tool, XAMN (version 3.2), where the Android file system was reconstructed. Several digital forensic analysis techniques were then used to locate files and artifacts related to MIoT smartphone applications. These techniques included: string searching, text filtering, and browsing the respective file systems.

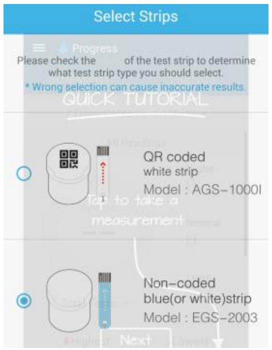
The scope of this research is restricted in the following ways. The experiment was conducted in the United States (US) using devices that contain network software for carrier providers in the US. The MIoT devices used in the experiment were acquired through the manufacturer’s US-based website. The experiment focused on a specific version of the Android and iOS operating systems, specific versions of the MIoT smartphone applications, and a specific version of XRY and XAMN. Due to tool limitation, Android was the sole operating system used for the hijacking smartphone. The experiment was executed only once, on each victim smartphone, with only one hijacking smartphone device. It should also be noted that the primary researcher was both a participant and a researcher in the experiment.

4. Results and Analysis

This section presents an analysis of the hijacker and the MIoT device pairing, as well as the results of the manual and smartphone examinations.

4.1 Pairing of MIoT Devices and Hijacker

At a high-level, two of the MIoT devices (the glucometer and the oximeter) were successfully added to the hijacker’s profile. This holds true for both the Android and iOS profiles. Several observations were made during this pairing process. From the perspective of the glucometer, a hijacker can ‘pair’ the device with their smartphone. First, the hijacker is prompted to scan either white QR-coded or blue non-coded test strips (Figure 1), before they can add the glucometer to their profile. To bypass the above selection, a hijacker can select the “non-coded strip” selection option. Next, the hijacker is notified to confirm that the device is powered-on and prompts the hijacker to turn on Bluetooth capabilities on their smartphone. The hijacker can then scan for nearby devices and selects the glucometer by selecting the device name “BG5xxxxxx”, in the Bluetooth menu. This allows the hijacker to pair and connect their smartphone to the glucometer.

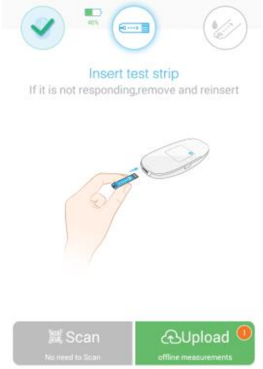

Select Strips

Please check the of the test strip to determine what test strip type you should select.

* Wrong selection can cause inaccurate results.

Figure 1: Glucometer Strip Selection

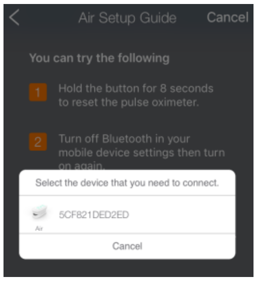
After the pairing was successful, the hijacker is presented with a screen, as shown in Figure 2, prompting the upload of offline readings from the victim glucometer.


Insert test strip

If it is not responding,remove and reinsert

Figure 2: Glucometer Offline Readings

In terms of the oximeter, the hijacker is again notified to confirm that the device is powered-on and prompted to turn on Bluetooth capabilities on their smartphone. The device setup guide then searches for an oximeter nearby and then prompts the hijacker to select any devices that have been found (Figure 3). A Bluetooth connection between the hijacker and the oximeter is established at this point.



Air Setup Guide Cancel

You can try the following

1 Hold the button for 8 seconds to reset the pulse oximeter.

2 Turn off Bluetooth in your mobile device settings then turn on again.

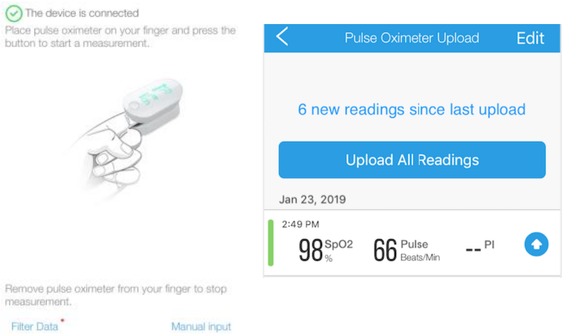
Select the device that you need to connect.

5CF821DED2ED
Air

Cancel

Figure 3: Oximeter Device Selection

The hijacker is then provided with the option to take a new reading with the device or to upload any offline data through the 'Filter Data' option, as shown in Figure 4. Selecting this option provides the hijacker with a list of all the offline readings that are currently stored on the oximeter device.



The device is connected

Place pulse oximeter on your finger and press the button to start a measurement.

Remove pulse oximeter from your finger to stop measurement.

Filter Data Manual input

Pulse Oximeter Upload Edit

6 new readings since last upload

Upload All Readings

Jan 23, 2019

2:49 PM
98 SpO2 % 66 Pulse Beats/Min -- PI

Figure 4: Oximeter Offline Readings

The scale is the only MIoT device requiring the hijacker to interact with the device physically, in order to 'pair' with the hijacker's smartphone. For this device, the hijacker is required to push a button at the front of the scale to turn on 'broadcast mode'. After the device is placed into this mode, the hijacker is notified that a scale has been detected and the smartphone application configures the device for use. However, even after the hijacker is notified that the scale has been successfully paired with the smartphone, they are not prompted to upload any offline readings from the device.

4.2 Hijack Device Examination

A summary of MIoT device recordings recovered from the hijacker smartphone is available in Table 3 – Summary of Results. Several observations are derivable from these results. The analysis of the hijacker smartphone confirmed initial observations: the smartphone did not 'pair' with the scale. As a result, no data from this MIoT device was visible on the hijacker's smartphone interface, nor recovered from the smartphone's memory using the forensic toolkit.

However, an examination of the hijacker's smartphone revealed that medical information from the glucometer and oximeter devices was recoverable. This information included both Android and iOS application profiles. Depending on the MIoT device, this information is visible in either the smartphone's interface or the forensic extraction of the smartphone's internal memory. This information confirms the initial assumption that the hijacker's smartphone successfully 'paired' with the glucometer and oximeter devices.

OS	**Device/Reading Value**	**Day 1**	**Day 2**	**Day 3**
Android	Glucometer:			
	Blood Sugar	M	M	M
	Timestamp	M	M	M
	Oximeter:			
	Oxygen Level	✓	✓	✓
	Pulse	✓	✓	✓
	Timestamp	✓	✓	✓
	Scale:			
	Weight	X	X	X
	Body Fat	X	X	X
	Body Water	X	X	X
	Pulse	X	X	X
	Bone Weight	X	X	X
	Muscle Weight	X	X	X
	BMI	X	X	X
	Timestamp	X	X	X
iOS	Glucometer:			
	Blood Sugar	M	M	M
	Timestamp	M	M	M
	Oximeter:			
	Oxygen Level	✓	✓	✓
	Pulse	✓	✓	✓
	Timestamp	✓	✓	✓
	Scale:			
	Weight	X	X	X
	Body Fat	X	X	X
	Body Water	X	X	X
	Pulse	X	X	X
	Bone Weight	X	X	X
	Muscle Weight	X	X	X
	BMI	X	X	X
	Timestamp	X	X	X

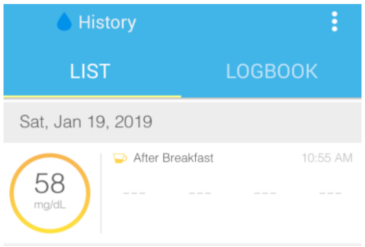
Key: ✓ = Recovered using manual and forensic examination; M = Recovered using manual examination only; X = Not recovered using manual or forensic examination

Table 3: Summary of Results

4.2.1 Manual Examination

A manual examination of the smartphone revealed that the hijacker's version of the Gluco-Smart and MyVitals applications contained medical information (i.e., device readings). Figure 5 – Glucometer Manual Examination and Figure 6 – Oximeter Manual Examination present the results of the manual examination of these applications.

From the perspective of the glucometer, a hijacker can potentially, view readings taken using the device in the Gluco-Smart application. The information recovered from the victim's glucometer includes the victim's blood sugar level, along with the date and time of the acquired reading.



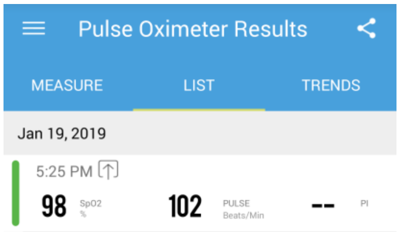
Sat, Jan 19, 2019

After Breakfast 10:55 AM

58 mg/dL --- --- --- ---

Figure 5: Glucometer Manual Examination

Similarly, the manual examination of the MyVitals application interface revealed that a hijacker could view device readings from the oximeter device. This application reports a victim's pulse rate in beats per minute, oxygen level, and each readings timestamp.



Jan 19, 2019

5:25 PM

98 SpO2 % 102 PULSE Beats/Min -- PI

Figure 6: Oximeter Manual Examination

4.2.2 Forensic Image Examination

The analysis of the Android memory dumps revealed a variety of artifacts related to the Gluco-Smart and MyVitals applications. Artifacts related to these applications can are located in different subfolders under the location `/data/ data/` in the Android filesystem [16]. The location of specific artifacts varies depending on the application under investigation. Unless noted, all timestamps recovered from the Android forensic extractions are recorded as epoch timestamps.

The MyVitals application creates a folder called `iHealthMyVitals.V2`, which is stored in the following file path `/data/data/iHealthMyVitals.V2`. This folder contains artifacts related to the victim's use of the Air pulse oximeter. Within the high-level folder, there is a `Databases` subfolder. This contains various SQLite databases of potential interest. A database called `androidNin.db` contains fifty-seven (57) tables, including three tables of data relevant to the victim's oximeter and the hijacker. A table called `TB_Device` contains information about the oximeter device, which has been subject to interactions with the hijacker smartphone. Information regarding the oximeter that is available in this table includes the model number, the firmware version, and the physical MAC address of the device. A second table, which contains information related to the victim oximeter, is called `TB_Spo2OfflineResult`. This table contains the actual hijacked readings that the victim undertook using the oximeter. Along with the oximeter reading result, a hijacker can also obtain the date and time the reading was acquired, the timezone where the reading was taken, along with the MAC address of the oximeter used by the victim.

A final table of interest is called `TB_UserInfo`. This table contains information about the hijacker and is likely to be of interest to an incident handler or forensic investigator. This table contains the date of birth, gender, height, and weight information as provided by the hijacker when the hijacker' MyVitals profile was created. In addition to the information in the database file, relevant information was also found in several Extensible Markup Language (XML) files. These XML files are stored in a subfolder called `shared_prefs`, under the iHealthMyVitals.V2 parent folder. Within the `shared_prefs`, subfolder the following files and information can be recovered, related to the hijacker, their victim, and the interactions between their devices:

- `historyTime.xml` – contains the MAC address and timestamp information regarding the last interaction between the hijacker's smartphone device and MIoT device.
- `saveUserIDs.xml` – contains the email addresses used by the hijacker to register their account with the iHealth service.
- `saveDeviceId.xml` – contains a list of the MIoT devices, which the hijacker has successfully accessed using this specific account. This includes the MAC address of the victim devices.
- `sp_connect_times_file.xml` – contains the number of times that the hijacker's account has been used to access a specific MIoT device. This information includes the MAC address of the device and the number of previous connections.

- sp_user_region_host_info.xml – contains information about the hijacker including their application account access token, a hash of their password, along with the cloud host used to access the iHealth services.
- device_id.txt.xml – contains the device Universal Unique Identifier (UUID) of the hijacker's smartphone.

The Gluco-Smart application creates a folder called jiuan.androidBg.start, which is stored under the location /data/data/jiuan.androidBg.start. This folder contains artifacts related to the victim's use of the Smart glucometer. While a database called androidBG.db was recovered from a subfolder called Database within the high-level application folder, this database was encrypted. An analysis of the other database files in the subfolder did not reveal any information about the hijacker or the victim devices. However, various XML files stored in the shared_prefs subfolder provide detailed information about the hijacker and their activities. The following files and information can are recoverable from this subfolder:

- USER_INFO.xml – the username (as an email address) and the smartphones' UUID that is used to connect to the glucometer device.
- sp_user_region_host_info.xml – contains the hijacker's email information, along with the host used to access the iHealth services.
- sp_last_update_TS.xml – contains the MAC address and timestamp information regarding the last interaction between the hijacker's smartphone device and MIoT device, whose MAC address is listed.
- sp_connect_times_file.xml – contains the number of times that the hijacker's account has been used to access a specific MIoT device. This information includes the MAC address of the device and the number of previous connections.
- saveDeviceIdTS.xml – contains a list of the MIoT devices, which the hijacker has successfully accessed using this specific account. This includes the MAC address of the victim devices.
- device_id.txt.xml – contains the device UUID of the hijacker's smartphone.

4.3 Analysis Summary and Limitations

The results described above can be used to provide answers to the research questions proposed in Section One. First, the analysis of the hijacker smartphone revealed that the hijacking attacks on the victim MIoT devices resulted in recoverable residual data on the hijacker's smartphone.

Second, the manual and forensic analysis of the hijacker smartphone revealed that victim MIoT device readings are recoverable through a hijacking attack. The results of the manual examination have shown that the hijacker's smartphone application contains readings from two (glucometer and oximeter) out of the three victims MIoT devices. Moreover, the results from the forensic examination of the hijacker's smartphone revealed that device readings from the oximeter are recoverable from databases stored on the smartphone.

Third, in addition to recovering device readings, the hijacker smartphone also contains a variety of metadata related to the glucometer and oximeter. This metadata includes device model numbers, firmware versions, and MAC address information. While this information is recoverable using a forensic extraction of the hijacker's smartphone, if the same smartphone has been 'rooted,' a hijacker could potentially recover this information using tools freely available on the Internet at no cost. This information would be useful to an attacker interested in potentially causing a denial of service attack against MIoT devices [43].

While the results of the manual and forensic examination of the hijacker's smartphone revealed information about the victim MIoT devices, minimal information about the victim is recovered from this experiment. However, previous research establishes that an attacker can identify high-level device data patterns based on residual data generated from a variety of smartphone applications [19]. As a result, if an attacker combines information about the victim from other smartphone applications, coupled with the intelligence gathered from the MIoT devices, high-level data patterns are a possibility.

The overall analysis of the data partially supports the hypothesis that MIoT devices are susceptive to hijacking attacks through their accompanying smartphone applications. The hypothesis statement is true for two out of the three MIoT devices evaluated in this research. This statement holds true for both the Android and iOS application profiles on these devices. The data intercepted by a potential attacker could be used to commit further attacks or augment user profile development.

While the analysis demonstrates that it is possible to launch a hijacking attack against a MIoT device, the following assumptions and limitations must be acknowledged. First, it is assumed that the hijacker is within proximity to the MIoT device. This is required to maintain a Bluetooth connection. However, due to the mobility feature of MIoT devices, this is not implausible as victims may use these devices in public places such as airports, libraries and coffee shops.

Second, this method of attack is relevant while the device manufacturer does not implement a verification mechanism, such as the approach used in the scale. If physical access to the MIoT device is needed to enable a feature or to push a button, then the hijacking attack is invalided using the proposed attack model. Third, the attack model is successful because the victim is not prompted to confirm if a particular smartphone can connect and receive information from the MIoT device. If a manufacturer enables such a feature, the victim will be prompted that a malicious hijacker is attempting to connect to a device and the hijacker’s smartphone does not successfully ‘pair’ with the MIoT device.

5. Conclusion and Future Work

The amalgamation of the Internet of things (IoT) devices into medical scenarios creates an atmosphere that is conducive to a variety of attacks. The results of this proof of concept research support the hypothesis that medical IoT devices are susceptible to hijacking attacks. The data demonstrates that it is possible to launch a successful hijacking attack against a Medical IoT (MIoT) device. This attack, potentially, allows an attacker to gather information about the MIoT device user, as well as the device itself. This intelligence could then be combined with other smartphone data to develop detailed profiles about the individual, including the identification of potential health issues. Moreover, the intelligence gathered from a MIoT device could also be used to launch denial of service attacks against similar devices. Such attacks on medical devices can be especially problematic in an emergency scenario.

Future research will examine several key areas. This research will expand to include diverse MIoT devices and associated smartphone applications. The focus of this experiment is to evaluate the results from this initial investigation on a larger scale. Further research also needs to examine MIoT smartphone applications from the perspective of multiple operating systems. Future work will explore the idea of automating the attacks described in this paper, along with other vulnerabilities in MIoT devices. This automation then allows for the development of a test environment that will assist with the interrogation of medical devices and the development of potential secure mitigation strategies.

6. Acknowledgments

Financial support for this research is provided by the Nebraska Research Initiative (NRI).

7. References

[1] Soroush, H., D. Arney, and J. Goldman, *Toward a Safe and Secure Medical Internet of Things.* IIC Journal of Innovation, 2016. **2**(1): p. 4-18.

[2] Pureswaran, V. and P. Brody, *Device democracy: Saving the future of the Internet of Things*. 2015.

[3] European Commission, *New EU rules to ensure safety of medical devices*. 2017, European Commission: Brussels, Belgium.

[4] PricewaterhouseCoopers, *The Wearable Life 2.0 - Connected living in a wearable world*. 2016.

[5] Hutton, L., B.A. Price, R. Kelly, C. McCormick, A.K. Bandara, T. Hatzakis, M. Meadows, and B. Nuseibeh, *Assessing the privacy of mhealth apps for self-tracking: heuristic evaluation approach.* JMIR mHealth and uHealth, 2018. **6**(10): p. e185.

[6] Hu, F., D. Xie, and S. Shen. *On the application of the internet of things in the field of medical and health care*. in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. 2013. IEEE.

[7] Qualcomm Life. *The Power of Medical Device Data*. 2015. http://blog.capsuletech.com/infographic-the-power-of-medical-device-data. Date of Last Access: 1st May, 2019.

[8] Qualcomm Life. *Successfully Navigating Mobile Challenges in the Health Care Landscape*. 2016. http://www.qualcommlife.com/images/pdf/white_papers/QCL_whitepaper_mobile_02.28.15_web.pdf. Date of Last Access: 1st May, 2019.

[9] Stanford Medicine, *Stanford Medicine Health Trends Report – Harnessing the Power of Data in Health*. 2017.

[10] United States Government, *The Health Insurance Portability and Accountability Act, Pub.L. 104-191*, United States Government, Editor. 1996.

[11] United States Department of Health & Human Services, *Security Standards: Administrative Safeguards*. 2007.

[12] United States Government, *Code of Federal Regulations - Title 45 - Public Welfare.* 2007: p. 738.

[13] Azfar, A., K.-K.R. Choo, and L. Liu, *Forensic Taxonomy of Popular Android mHealth Apps*, in *21st Americas Conference on Information Systems*. 2015: Puerto Rico, USA.

[14] Grispos, G., W.B. Glisson, and T. Storer, *Using Smartphones as a Proxy for Forensic Evidence Contained in Cloud Storage Services*, in *46th Hawaii International Conference on System Sciences*. 2013: Hawaii, USA.

[15] Grispos, G., W.B. Glisson, and T. Storer, *Recovering Residual Forensic Data from Smartphone Interactions with Cloud Storage Providers*, in *The Cloud Security Ecosystem*, K.-K.R. Choo and R. Ko, Editors. 2015, Syngress: Boston. p. 347-382.

[16] Hoog, A., *Android Forensics: Investigation, Analysis and Mobile Security for Google Android*. 2011: Syngress.

[17] Levinson, A., B. Stackpole, and D. Johnson, *Third Party Application Forensics on Apple Mobile Devices*, in *44th Hawaii International Conference on System Sciences*. 2011: Hawaii, USA.

[18] Grispos, G., W. Glisson, and P. Cooper. *A Bleeding Digital Heart: Identifying Residual Data Generation from Smartphone Applications Interacting with Medical Devices*. in *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019. Hawaii, USA.

[19] Grispos, G., W.B. Glisson, J.H. Pardue, and M. Dickson. *Identifying User Behavior from Residual Data in Cloud-based Synchronized Apps*. in *Proceedings of the Conference for Information Systems Applied Research*. 2014. Baltimore, MA, USA.

[20] Wood, D., N. Apthorpe, and N. Feamster. *Cleartext data transmissions in consumer iot medical devices*. in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. 2017. ACM.

[21] Classen, J., D. Wegemer, P. Patras, T. Spink, and M. Hollick, *Anatomy of a vulnerable fitness tracking system: Dissecting the fitbit cloud, app, and firmware.* Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018. **2**(1): p. 5.

[22] Grispos, G., W.B. Glisson, and K.-K.R. Choo, *Medical Cyber-Physical Systems Development: A Forensics-Driven Approach*, in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*. 2017: Philadelphia, USA.

[23] McMillan, J., W.B. Glisson, and M. Bromby. *Investigating the Increase in Mobile Phone Evidence in Criminal Activities*. in *Hawaii International Conference on System Sciences (HICSS-46)*. 2013. Wailea, Hawaii.

[24] Berman, K., W.B. Glisson, and L.M. Glisson. *Investigating the Impact of Global Positioning System (GPS) Evidence in Court Cases*. in *Hawaii International Conference on System Sciences (HICSS-48)*. 2015. Hawaii, USA.

[25] Grispos, G., W.B. Glisson, and T. Storer. *Cloud Security Challenges: Investigating Policies, Standards, And Guidelines In A Fortune 500 Organization*. in *21st European Conference on Information Systems*. Utrecht, The Netherlands.

[26] Malasri, K. and L. Wang, *Securing Wireless Implantable Devices for Healthcare: Ideas and Challenges.* IEEE Communications Magazine, 2009. **47**(7): p. 74-80.

[27] Glisson, W.B., T. Andel, T. McDonald, M. Jacobs, M. Campbell, and J. Mayr, *Compromising a Medical Mannequin*, in *21st Americas Conference on Information Systems*. 2015: Puerto Rico, USA.

[28] Li, C., M. Zhang, A. Raghunathan, and N.K. Jha, *Attacking and Defending a Diabetes Therapy System*, in *Security and Privacy for Implantable Medical Devices*, W. Burleson and S. Carrara, Editors. 2014, Springer. p. 175-193.

[29] Baker, S.B., W. Xiang, and I. Atkinson, *Internet of things for smart healthcare: Technologies, challenges, and opportunities.* IEEE Access, 2017. **5**: p. 26521-26544.

[30] Demidowich, A.P., K. Lu, R. Tamler, and Z. Bloomgarden, *An evaluation of diabetes self-management applications for Android smartphones.* Journal of telemedicine and telecare, 2012. **18**(4): p. 235-238.

[31] Dimitrov, D.V., *Medical internet of things and big data in healthcare.* Healthcare informatics research, 2016. **22**(3): p. 156-163.

[32] Rao, A., P. Hou, T. Golnik, J. Flaherty, and S. Vu, *Evolution of data management tools for managing self-monitoring of blood glucose results: a survey of iPhone applications.* Journal of diabetes science and technology, 2010. **4**(4): p. 949-957.

[33] Vashist, S., E. Schneider, and J. Luong, *Commercial smartphone-based devices and smart applications for personalized healthcare monitoring and management.* Diagnostics, 2014. **4**(3): p. 104-128.

[34] Sun, W., Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, *Security and privacy in the medical internet of things: a review.* Security and Communication Networks, 2018.

[35] Williams, P.A. and V. McCauley. *Always connected: The security challenges of the healthcare Internet of Things*. in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 2016. IEEE.

[36] Aliasgari, M., M. Black, and N. Yadav. *Security Vulnerabilities in Mobile Health Applications*. in *2018 IEEE Conference on Application, Information and Network Security (AINS)*. 2018. IEEE.

[37] Fereidooni, H., T. Frassetto, M. Miettinen, A.-R. Sadeghi, and M. Conti. *Fitness trackers: fit for health but unfit for security and privacy*. in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. 2017. IEEE.

[38] Lotfy, K. and M.L. Hale. *Assessing pairing and data exchange mechanism security in the wearable Internet of Things*. in *2016 IEEE International Conference on Mobile Services (MS)*. 2016. IEEE.

[39] Siddiqi, M., V. Sivaraman, and S. Jha. *Timestamp integrity in wearable healthcare devices*. in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. 2016. IEEE.

[40] Oates, B.J., *Researching Information Systems and Computing*. 2005: Sage.

[41] Statista. *Smartphone shipments by vendor worldwide from 4th quarter 2009 to 1st quarter 2019 (in million units).* 2019. https://www.statista.com/statistics/271490/quarterly-global-smartphone-shipments-by-vendor/. Date of Last Access: 1st May, 2019.

[42] Grispos, G., T. Storer, and W.B. Glisson, *A Comparison of Forensic Evidence Recovery Techniques for a Windows Mobile Smart Phone.* Digital Investigation, 2011. **8**(1): p. 23-36.

[43] Lin, G. and G. Noubir, *On link layer denial of service in data wireless LANs.* Wireless Communications and Mobile Computing, 2005. **5**(3): p. 273-284.