[[1]](#footnote-1)

Privacy and Security Issues in Wearable devices

Survey on MIFIT and Recommendation

Saksham Yadav ([syadav1@kent.edu)](mailto:syadav@kent.edu)), Abhishek Gupta ([agupta12@kent.edu)](mailto:agupta12@kent.edu))

Kambiz Ghazinour (kghazino@kent.edu), Kent State University, Kent Ohio, USA

*Abstract*— the privacy of an individual is of paramount importance. The person or individual must have the power of what information he/she wants to share and what information he wants to not share with anybody. In this paper we will discuss about the wearable technology and the privacy issues in these devices. We will also suggest and recommend on how the user can be educated about these vulnerabilities and suggest some solution on how the individual can safeguard his/her interest.

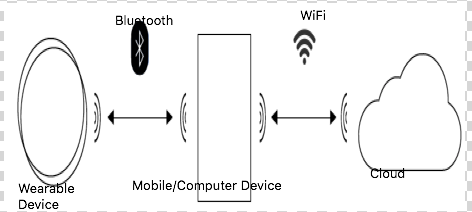
Keywords: Wearable technology; FITBIT; MIFIT; Privacy; Security

# **INTRODUCTION**

Wearable devices are fast growing industry in the field of mobile technology. The ability to communicate with the phone and the capabilities to exploit the IOT technology and integrate with the mobile phones have made the contribution to the growth of Wearable technology. With the significant improvement of battery life of the Wearable devices have made huge progress in improvement of Wearable devices. More fitness devices have come to market which help the person to track of his daily activities. Common among them is the FITBIT, MIFIT, and Samsung Gear etc. In this paper we will talk about the general architecture of Wearable technology, BTLE, literature Review about FITFIT, Analysis of MIFIT. The privacy issues in Wearable technology. The possible solutions that are currently available. We end up with recommendation and future scope and future work.

**1: ARCHITECTURE OF WEARABLE TECHNOLOGY.**

The Figure 1 shows the general architecture of Wearable technology communication [1][2]. It contains the wearable device and associated mobile device. These two are communicating through the BTLE protocol. The information about you stored in the phone collected by Wearable device is stored in the mobile phone. This information can be stored in the computer also by transferring the data from phone to the computer and Synced. The Wearable device manufacture also provide option to store the information collected by them to the online cloud maintained by them. Some will use third party servers to store the data. The application setting will allow you to regularly sync your data from phone into the servers hosted by them. Some companies also provide the analysis of the data and provide much more individual feedback about you with extra money.

******

## Figure 1: Architecture of Wearable Device communication.

## **2: BTLE PROTOCOL**

Bluetooth Smart (Low Energy) technology [3][4] has contributed hugely to the development of the Wearable devices. It is also called as BLUETOOTH 4.0. This was mainly built for the communication of devices over the Internet of things. This was also beneficial to the development of wearable devices since it also exploited the service of communication between the sensors. All it needed was the Bluetooth device to communicate with phone. The BTLE is similar to the regular Bluetooth protocol but it consumes significantly less energy than the regular protocol. It uses 128-bit AES with Counter Mode CBS-Mac encryption. The range of the BTLE is similar to normal BTLE protocol and is around and Maximum of 100 meters

The data can be send within 3 milliseconds. One of the advantages of using this protocol is you don’t need to use the slaves which was mandatory in the classical Bluetooth protocol. The peak power consumption is 0.01 W to 0.5 W compared to classic Bluetooth protocol. Even the peak current consumption is less than 15mA compared to that of 30mA of the classic Bluetooth protocol. The reason why BTLE consumes such less power is because this protocol doesn’t support the voice and video capability.

**3: LITERATURE REVIEW: FITBIT**

While we were researching about the wearable devices which concentrated on the fitness bands we found a research done by the MIT students about the security and privacy issues

Of the FITBIT [5] device. In that paper they talked about the various security issues and privacy threats that FITBIT users will encounter. The paper in detail explained about the process they followed in order to check the information that FITBIT tell it will collect and what it actually collect during the process. It also talks about the security issues by using FITBIT and they have also proposed how the security of the FITBIT can be improved by rewriting the software of FITBIT to protect from the vulnerabilities of the device. We were only interested in the privacy issues in our project as we were interested about the privacy violation of the fitness bands in the market in general.

The majority of privacy violation of the FITBIT occurs during the pairing process. During the pairing process the FITBIT collects the information of other FITBIT near the coverage area of it. The information will be stored in the phone according to their findings of the paper. One of the major concerns in Wearable devices is information about the person stored and what will they do with that collected information. Since the application developers will always be in Dilemma of which law of framework. These laws will differ from country to country. So the application developers will always try to ignore the privacy aspect of the application. Developers cannot change the functionalities that they are providing according to rules of every country. So the companies will always try to neglect the issues until someone raise the issue.

**4: ANALYSIS OF MIFIT APPLICATION**

**MIFIT** is the wearable device that is similar to FITBIT application developed by Xiaomi. It is one of the famous handset manufacturer in Asian Market. The reason we chose to analyze this because there was huge uproar in India about data being collected by the phone and where the information is stored and even to the extent of sharing the information to the Chinese government even of the Indian customers the phone had. They had issued a public statement that the customers [6] need not be worried about the issue.

They have a flagship product developed called as MI Band which provides functionalities such as monitoring the distance travelled during the particular day and also the sleep monitor. We cannot see the results in the MI Band itself as it does not have GUI of itself but we need to install the application (MIFIT) that they have provided in your phone to see the results of your activities. So we decided to analyze the device and what information it collects. Our first step in our attempt was to analyze the privacy policy of the device and the application that is being installed in the phone. We are concentrating on the application of android phone. We have checked with Moto 3rd Gen, Nexus 6 and Xiaomi (Mi4i) phone for testing purpose.

Our main goal is to study what MIFIT collects, what is being stored locally and what information is being send to the server. Map out any security issues and check if that information being sent in any way compromise the security of the individual.

First step in our attempt is to get the application (MIFIT).We got the .apk file from the application using the APK extractor [7]. The .apk is a zipped and signed archive of the application and associated linked resources of the application. Apk extractor provides us the option to E-mail extracted application which can later use to analyze on the computer.

Android machine readable code is .dalvik code. We got the .dalvik code of the MIFIT application from the apk extractor. Machine readable code is difficult to understand so we needed some tools to convert the machine application to human under stable source code.

We have used dex2jar [8] to covert the machine readable .dalvik file to the .jar files which can later be converted to java files from the class files we obtained. We will not get complete source code from this application but we will get enough knowledge of what the application does and what is being send. So we looked on how to access the Bluetooth [9] data and we were unsuccessful in our efforts.

Second step we analyzed the log files that is being send to the server and the log files that is being stored in our phone. We try to map put the source code and application logs on what information is being sent from the device and phone and from phone to the sever and what kind of information is being sent to the server during the synchronization process.

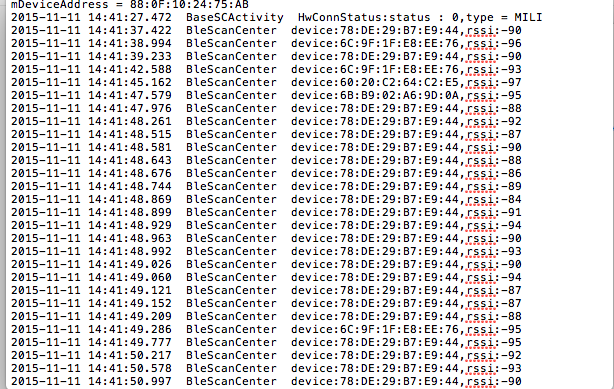


Figure 2: Log Files obtained during the synchronization process of the MIFIT and android phone.

Figure 2 shows the log file extract we had collected using the pairing process of the phone and the MIFIT application. During the process of pairing it collects the information of nearby devices. It is standard to scan for devices during Bluetooth discovery but we have also noted that this information also being stored in the computer and stored in the log files. We also noticed that information being transmitted to the server. We are not sure what other information is also being send to the server.

From examining further log statements, we were able to discern much of how the Bluetooth protocol works. The device authenticates with the app by computing a MAC over random bits, using a CBC-MAC with the XTEA block cipher. So if once the device is synced and given permission to control to the phone via MIFIT and it is synced with device. We can switch on the Bluetooth from mobile phone and access the phone without entering the phone passcode. This would pose a serious security risk if a person misplaces the phone.

The next step in analyzing what information and where the information is being send. We have used Burp Suite to check out what information is being sent over the network by MIFIT. Figure 3 shows the analysis from the Burp Suite [10].

From the process we were able to observe all the communication between the mobile device and server. We tried to look what kind of security measures or protocols have been used during the process. The answer was quite surprising. The communication was done through only HTTP protocol and the much more secure SSL/TSL protocols were not used.

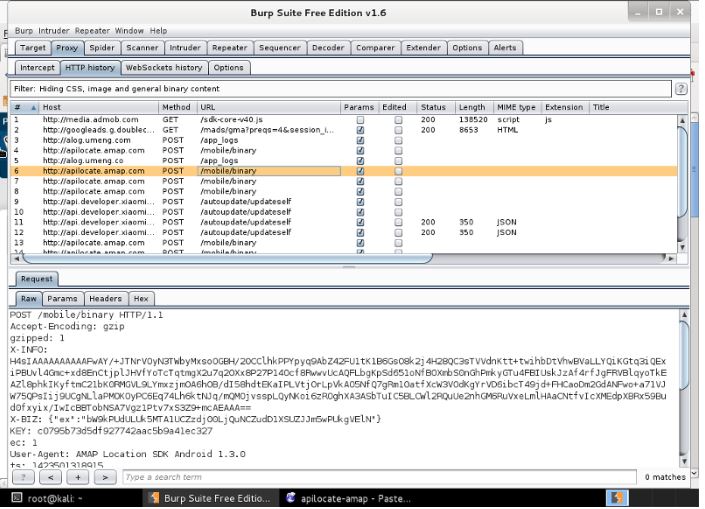


Figure 3: Communication captured during synchronization process between the computer and to the server.

MIFIT allows us to store the information and sync the data with the Google Fit services offered to the android users. We tried to analyze that data is being sent to which servers. We found that it is sending data to a total of 15 servers among which we found three servers which was neither related to the Xiaomi nor with google servers for google fit communication. We tried to look at the privacy policies it is did not stated anywhere that it will store or send the information to third party server. However, they can sell the customer information for the purpose of research to its subsidiary companies. However, there is no clear mentioning of where the information is being stored. They have said that it will not share the information without the being taking consent. The privacy policy is mentioned in their online website. The privacy policy of the MIFIT which we get while is very vague and since the MIFIT band we brought from India. We don’t know how the privacy laws vary from country to country regarding buying a product from one country and using it in another country. However, in the privacy policy they have mentioned they will follow local jurisdiction laws and is valid under the local jurisdiction. However, the jurisdiction term is very grey area and can be interpreted differently.

There was also other important information we noticed while we were analyzing the services. We tried to switch off the location discovery feature for the device however it somehow uses information of GSM and other information to calculate the location of the customer which we were able to identify from the hex dump stored in the location cache of the application stored in the application without much significant effort. In the log files we found that there is information stored in mi. health database apparently we were not able to retrieve it. It might be the limitation of reverse engineering we have followed or it might be dynamically linked so we are not able to link to the java files we got. Figure 4 shows the encrypted location information from the log files.

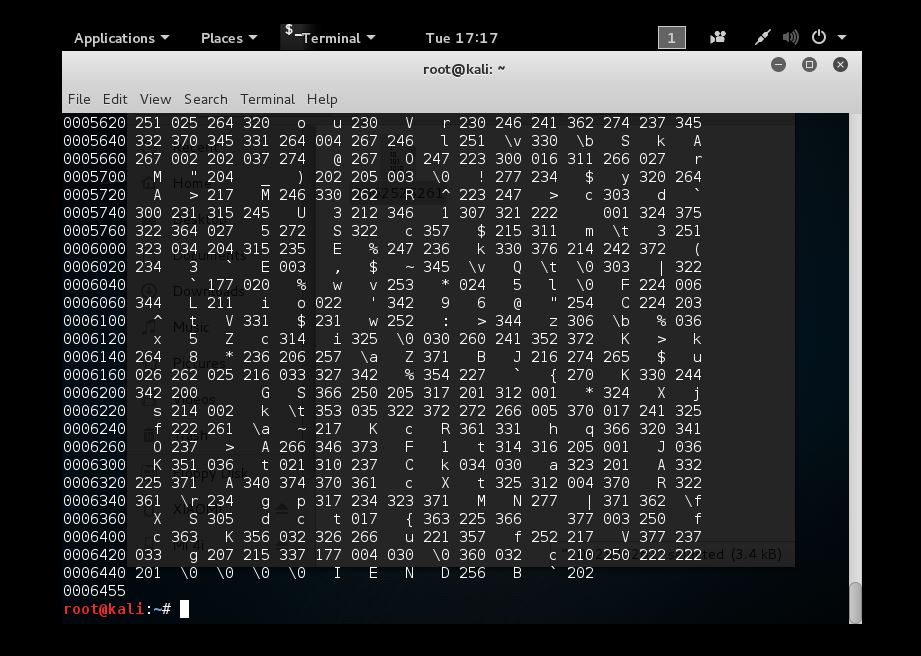


Figure 4: Encrypted Log files of location of the application

The application uses the Baidu service for the location based analysis. The company provides the special analysis based on geo location data for research purpose. But we didn’t found any information regarding it Xiaomi application. We tried to analyze the *AndroidManifest.xml* to find out what function are defined and there functionality. We also analyzed the class files and try to map put what functions and encoding is done by the application. We also found out the information of phone is stored in the log files. We found that information about the phone is encoded shown below

Mod=GT-GP916Q&sv=4.2.1&nt=WIFI&np=TIM

It contains the phone serial number and version and protocol used in transport layer communication. It uses TIM (TCP-IP inverse multiplexing.

We also tried to look at the Moto 360 and associated fitness application. But we didn’t not enough time to look at the issues that a Moto 360 can face. But we were able to map out some common issues that will affect the user privacy issue. In our next section we will describe about how a person can prevent unauthorized access by application. If we don’t like some permissions required by the application but you still need to use the application in a secure way and control what information is being send.

**5: POSSIBLE SOLUTION**

Since we cannot control the way what we share to the application to the company. We need a workaround so that people can be careful about what amount of information they want to share and what they don’t want to share. There are some workarounds we have found in the Internet. We are presenting some solution that we are available in the market.

# **A: XPRIVACY**

Xprivacy [10] [11] is a program on android (only rooted ROM’s e.g. cyanogen mod [12]) platform which helps in getting control of all the apps that are installed on the device. Basic idea behind this application is that while installing a new application on mobile device we need to agree many terms and conditions whether we want to accept or not in order to get services from that app. But this app can restrict or block several queries and activities even if there was permission granted at the time of installation.

Xprivacy takes very simple steps to be installed, all that is required is a rooted device. once installed on the device it uses several techniques to stop leaking privacy sensitive information (data) from the device, by restricting the categories (location, contacts) of data that the app can access on the device, it will have sent duplicate(fake) data to the application or in some cases don’t at all sent any data, E.g. - If “Snapchat” [13] application tries to access the contact list on the phone then Xprivacy will sent it an empty contact list by restricting applications access to original contacts. Good thing about this application is that it doesn’t block or revoke any permission that the application ask for hence the application can keep on running as before without crashing (force close).

There is still a big disadvantage of using xprivacy that it cannot stop other application from accessing internet or SD cards this is because access restriction to these services can only be done by revoking access permission from the application. What xprivacy can do to come out of this problem is to fake an offline mode but still some applications will try to access internet resulting in error or app crash. But it can allow the access to data in order to resolve the app crash problem. Xprivacy provides feature for on/off switch for all the imposed restriction on all the apps.

Once Xprivacy is installed on the device then all the new application that will be installed will not be able to access any data category, by this feature it helps to prevent any privacy sensitive data to be leaked. After the new application is installed, then Xprivacy will ask what all data category we want to allow access to the app.

Xprivacy can be installed in almost all the operating systems but there are still some problems with that as it still in its development phase for e.g.- it cannot be installed on a MIUI OS which is a android based OS customized by a Chinese company XIAOMI [2]

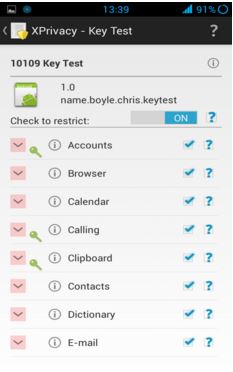


Figure 5: Categories of data an application can restrict access

The problem relates to the location based privacy. Nowadays mobile applications are dependent on the location of the application user to continue with the services. Sometimes, the services are the integral part of the functionality of the application. For example, google maps, in which the user cannot be guided by the exact location or coordinates. However, there are few applications where the exact location of the user is not mandatory to provide the services to the user. For example, when the user wants to know about the nearby restaurants and other nearby places to visit. At this point of time, the exact location of the user is not mandatory to suggest the restaurants.

**B: LOCATION HIDING**

The exactness of the location can breach the privacy of the application user. The identity of the individual person can be revealed tracing the exact location of the person. For example, the regular tracing of the location of the individual can reveal the daily routine of the person which in turn is stored at the MIFIT cloud. Consequently, challenging the personal identity of the individual.

In our research we found out that MI Fit collects the exact location of the person to collect and store the travel distance by the individual for each day to relate the distance walked to the health of the individual. However, there are ways to escape from sharing the exact location to the apps. Geographical Masking is one the process suggested in the paper [13]. The geographical masking uses the random approaches to assign the arbitrary nearby location of the individual. As a result, the exact location of the person is hidden and services of the person can also be made private. Thus, the exact location of the person can be replaced with the group of nearby locations and further during unmasking the exact location of the person is revealed on the local storage of the mobile phone.

The other suggested method for location hiding is giving controls to the user. The user should be given privileges to convert the representation of the parameters in the privacy preserved representation. For example, the data regarding the distance travelled can be captured and represented in the terms of the footsteps instead of representing it in the terms of miles. Consequently, the personal details of the user are hidden from the storage into the company’s server.

The other way is to anonymize the data location used by the mobile application [14]. The anonymized data does not reveal the exact information of the location of the user. Thus, hiding the few details from the exact location of the user of the mobile application can help in retaining the privacy of the user. Thus, the application should include the process to anonymize the location of the user up to certain level such that services should also not he hampered and the data can also be hidden from the exactness.

However, in the application we suggest that there should be some process of forgetting the previous location which is already covered by the user. The path which is already covered by the user while on the way should be formatted from the memory and should be converted into miles immediately to track the distance traveled. Thus, the exact path traveled by the application user will not be stored and it will only be stored in the format of miles. As there is the process of forgetfulness along with the covered path, the storage space will also be saved once it will be location is simultaneously converted into the miles. However, the processing of the application might Detroit as processing the location into miles simultaneously while on the way might consume the processor speed.

Here is the various application available in the android based mobile system [4]. These application allows the user to easily fake the location. One of the application is Fake GPS location Spoofer. The purpose of the application is to hide the exact location of the application user. Whenever an application tries to trace out the exact location of the user. The application pops up with option to block for the exact location of the user and further replace it with fake location. As a result, the user is able to use the services along with hiding the exact location of the user.

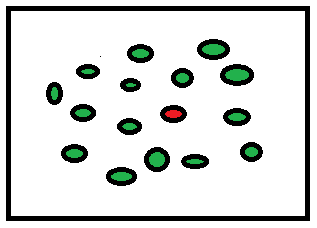


Fig 6: The cloaking of the exact location of user.

The other way to hide the exact location is by cloaking. In Fig. 6, the cloaking technique a region is defined by the application A (min) and A (max). A (min) refers to the minimum area required for cloaking the exact location of the user. Whereas A (max) is the maximum area required for cloaking. As shown in the figure the red point is exact location of the user. The red point is replaced by nearby location shown in green dots. Whenever the user search for specific location for example nearby restaurants, the restaurants of the nearby region are searched. Thus, resulting into hiding the exact location of the user and serving the purpose of the application.

**7: RECOMMENDATION**

We are proposing some recommendation that every developer across different operating systems and Wearable devices if followed would help in safeguarding the privacy of an individual.

Here are some of recommendation we are proposing listed below:

a) The user should be giving an option whether they want to share the information of their position that they have travelled. If they need their application of the person’s geo-location information is needed in order to calculate the steps, they have travelled. They can dynamically calculate the location by taking the GPRS information and delete all the information about individual travel history. Even if they store the information, it should not sync the data to the servers.

b) User should have what information they want to share the information to the server. The application of wearable devices should give an option in a radio buttons that this information is being sent to the server. If he doesn’t want his information to be stored in the server, the manufacturers should give an option for that also.

c) People should be informed about with whom the information is being shared with. The wearable device should not become a tool of surveillance for the government or some agencies.

4) The Wearable device should follow the XACML protocol that all the websites follow in order whether the match what information is being collected and what is written in their privacy policy.

5) The Standard Privacy Taxonomy should be followed and mentioned in some terms of GUI. The user should be mentioned how long their information is being stored and what information is being collected for what purpose.

6) While tracking of the wearable device during initial pairing process, the nearby devices which have their Bluetooth turned on their network discovery. We have noticed their information is also stored in the log files. We have noticed that information is also being sent to the server.

**8: FUTURE SCOPE**

We were not able to use any Network Sniffing tools which namely UBERTOOTH which would have helped us to snipe the live data of the communication that existed between the server and the mobile devices. If we know assuming based on our best guess that some information is being transferred

RE**FERENCE:**

### [1]The Architecture of Wearable Technology: Barry Peter Kirby, Ben Mosley. Report number: KS-WP-001, Affiliation: K Sharp Ltd, Research Gate Publication.

[2]Peer-to-peer Policy Management System for Wearable Mobile Devices UWolz, M. Massimi · DOI: 10.1109/ISWC.2003.1241421 Conference: Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on 10/2004.

[3]BTLE protocol :Design and Specification Available online at <http://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>

[4]“Bluetooth: With low energy comes low security,” M. Ryan, *iSEC Partners*Usenix,2013.[Online].Available:<https://www.usenix.org/conference/woot13/workshop-program/presentation/ryan>

[5]Security Analysis of Wearable Fitness Devices (Fit bit) Britt Cyr, Webb Horn, Daniela Miao, Michael Specter Massachusetts Institute of Technology Cambridge, Massachusetts, U.S.A.

[6]Article Regarding Xiaomi in TechCrunch article.Avalible online at <http://tech.firstpost.com/news-analysis/xiaomi-phones-steal-user-data-and-send-it-to-remote-servers-in-china-warns-indian-air-force-238673.html>

[7]“Apk extractor.” K. Heeseok, [Online]. Available:

<https://play.google.com/store/apps/details?id=net.sylark.apkextractor>

[8]“dex2jar: Tools to work with android. dex and java .class files.” [Online]. Available:

<https://code.google.com/p/dex2jar/>

[9] E. Chai, B. Deardorff, and C. Wu, “6858 hacking Bluetooth,” 2012. [Online].Available:

<http://css.csail.mit.edu/6.858/2012/projects/echai-bendorff-cathywu.pdf>

[10]Burp Suite Tool [Online]Available at <https://portswigger.net/burp/>

[11]Xprivacy -Copyright (c) 2013-2015 Marcel Bokhorst ([M66B](http://forum.xda-developers.com/member.php?u=2799345)) [online] <https://play.google.com/store/apps/details?id=biz.bokhorst.xprivacy.installer&hl=en>

[12]Xprivacy[online] available at

https://github.com /M66B/ XPrivacy

[13]Cyanogenmod [Online] Available at :<http://www.cyanogenmod.org/about>

[14]J. Zaddach and A. Costin, “Embedded devices security and firmware reverse engineering.” BH13US ’13 Las Vegas, USA

1. [↑](#footnote-ref-1)