Using Machine Learning Techniques to Track Individuals & Their Fitness Activities

Thomas Reichherzer, Mikayla Timm, Nathan Earley, and Nathaniel Reyes
Department of Computer Science, The University of West Florida
Pensacola, FL, 32514, USA
treichherzer@uwf.edu, (mpt4, npe5, njr9)@students.uwf.edu

Vimal Kumar
Department of Computer Science, The University of Waikato
Hamilton, New Zealand
vkumar@waikato.ac.nz

Abstract

The use of wearable devices for fitness and health tracking is on an upward curve with a range of devices now available from a number of manufacturers. The devices work with smart devices to exchange data via Bluetooth communication protocol. This paper presents the results of an initial study on the security and privacy weaknesses of wearable fitness devices. It discusses methods to 1) capture and process data sent from a wearable device to its paired smartphone during synchronization and 2) analyze the records to track individuals and make predictions. The data analysis methods use supervised machine-learning techniques to train a classifier for associating synchronization records with the individuals, their physical activities, and conditions under which they were performed. Results of the study show that the methods allow individuals and their activities to be tracked, both of which infringe on the privacy of the user. The paper also provides recommendations on improving the security of wearable devices based on the initial research results.

keywords: wearable fitness device; security; privacy; tracking; machine-learning techniques.

1 Introduction

Advancements in wireless communication, sensing and micro-electro-mechanical system (MEMS) technologies have resulted in a growing market of wearable medical devices. According to market research [9] worldwide revenue in the wearable medical devices market is expected to reach $2.3 billion by 2017, while more than 50 million devices will be shipped annually. A variety of wearable medical and fitness devices have come up, which collect continuous and discrete physiological data about the wearer's health. These devices are capable of collecting health related data such as blood pressure, electrocardiogram, heart rate, glucose level, etc.

Wearable devices are small and worn by their users throughout the day to collect data and store them on the device. However, because the devices have little storage capacity, periodically they must be paired with another device to offload the collected data. Many of them use a smartphone as a pairing device and both devices exchange data using Bluetooth as the wireless communication protocol. Once the data have been transported to a paired mobile device, it uploads them over an Internet connection to a Cloud service for further processing, storage, tracking, and visualization. The mobile device also visualizes the physiological data to the user.

The raw physiological data collected by wearable medical devices and exchanged between wearable and mobile device can be sensitive and private in nature containing personally identifiable information about the users including health information. For many manufacturers of wearables though, data throughput and energy consumption are major concerns and override often the security protocols. Sometimes, the primary security protocols rely on older technology and do not prevent unauthorized interception or corruption of the data.

We are interested in studying the vulnerabilities of the Bluetooth communication channel between a wearable device and its paired mobile smartphone to exploit its weaknesses and launch attacks on the confidentiality and privacy of data. For our initial study, we considered three different devices including the Fitbit, the Pebble Watch, and the Misfit Flash because they are widely available fitness-tracking devices that use the Bluetooth

4.0 communication protocol to exchange data with a smartphone. The smartphone runs an application from which users can trigger device synchronization with the wearable to collect the latest fitness data. Although we studied the security weaknesses of all three devices, we started with the Misfit Flash to build the customized tools for intercepting and analyzing the data.

Manufacturers of wearable devices integrate a range of different sensors such as dual axis thermal accelerometers, piezoelectric sensors, temperature sensor, etc. into a device to capture health and fitness related information about the user [8, 5]. However, the device collects and transmits abstract binary data from its sensor that require specific algorithms to interpret them and infer health or fitness-related information. The data processing algorithms are not run on the wearable device because of the limited computing capabilities and power availability. The wearable device instead transports them to a mobile device over the Bluetooth channel. Thus, information extracted from packets intercepted from the Bluetooth channel cannot be directly related to fitness or health-related information shown by a mobile device application. Instead, the binary data must be further processed, which is typically done by the manufacturer's Cloud services. Because the binary data transmitted over the Bluetooth channel are meaningless without the algorithms that process them, our project applied machine-learning techniques to learn an association between data captured during synchronization and the information shown to users on their mobile device such as number of steps, burned calories, and more.

The remainder of the paper describes our methods to collect, filter, and classify data from a wearable device to infer fitness activities of users. The paper presents results from our experiments as well as potential solutions to improve the security of wearables. Next, we describe previous work related to the security vulnerabilities of wearables and other Internet of Things (IoT) devices.

2 Related Work

The wearable technology and IoT industries are poised to become increasingly mainstream, shaping technology innovation to a wide range of applications in health and wellness. However, security challenges are a growing concern to consumers and the health care industry and improving security of wearables and wireless IoT devices and the awareness of security threats is a major area of research [11], [10], [2]. Some researchers have focused on the recording capabilities of wearables of nearby devices that could be exploited to track locations of users of a wearable device [4]. Our own work focuses on the application of machine-learning techniques to "make sense" of unencrypted binary data sent by a wearable to its mobile device.

Commercial and medical data is sensitive and personal. Hence, attackers should not be able to gain access to such data. Most commercial wearable devices such as Fitbit however use static MACs[4]. While static MACs have been used successfully in networks for years, in IoT and pervasive computing they represent a security gap, which can potentially become a vulnerability. Static MACs can be used to launch user correlation attacks to identify a unique data stream bound to a specific user. Static MACs further invite Bluesnarfing attacks.

3 Methods of Tracking & Detection

Our goal is to show that we are able to track users of wearable devices and distinguish between their activities. To accomplish this goal, we used several off-the-shelf hardware and software tools and combined them with simple python scripts to drive the data collection and perform filtering and other processing tasks. Our methodology consists of three steps: 1) trigger the wearable, 2) capture the data, 3) classify the data.

We start with triggering the wearable device to release its physiological data over the Bluetooth channel. Next, we capture the wireless data packets and extract the payload of individually transmitted wireless packets to reconstruct the complete record of data sent by the wearable device. Finally, we classify the extracted data using machine learning techniques to make meaningful inferences. The methods and tools used in each step are described below.

3.1 Triggering Data Release from a Wearable Device

To reverse-engineer the communication protocol we have collected and analyzed wireless packets exchanged between the wearable and mobile device during synchronization using the techniques described below. Following the analysis of the packets, we have successfully decoded the steps that trigger the release of data packets and reproduced them using gattool and a Python script as a driver program to automate the steps. The gattool software is a developer support tool that comes with the BlueZ Linux Bluetooth protocol stack. It generates Bluetooth packets based on our specification and sends them wirelessly to a nearby wearable device.

The steps to trigger the release can be summarized as follows:

- The wearable device begins advertising its basic device and connection information.
- The mobile application that wishes to connect to the wearable device sends a connection request to it.
- Based on the specific values and format of the data sent in the request, the wearable device starts releasing its data.

It should be noted that in this process the wearable device does not vet the source of the connection before allowing a connection to be established.

3.2 Data Collection & Pre-Processing

We use a low-cost USB Bluetooth Low Energy (LE) packet sniffer dongle that plugs into the USB port of a laptop or desktop PC running Windows 7 or higher. The accompanying sniffer software runs on the PC and captures data packets transmitted wirelessly by nearby devices. The software sends captured Bluetooth data packets to Wireshark for visualization and storage in the form of a pcap file. We use *tshark*, a command line tool capable of reading and processing pcap files, to convert individual Wireshark pcap files into text files for subsequent processing.

To collect data when the wearable and mobile device synchronize, we trigger the synchronization from a mobile device or through the steps outlined above. The Bluetooth sniffer and Wireshark collect the data. Once the Wireshark program shows empty data packets, indicating that the synchronization is complete, we save the captured data as a pcap file. Next, we run *tshark* to transform a pcap file to a text file. *tshark* combines the individual data packets captured by the Bluetooth Sniffer into a single transmission record that includes all messages exchanged between wearable and mobile device. It then saves the record in textual format in a new file. The produced text file is then run through a Python script that performs the following additional operations on the file: First, it identifies the beginning of each packet sent from the wearable device to the mobile device to extract its payload. Next, it combines the payload containing the sensed data into a new data record in binary format. After this, the script detects and removes redundant information from the binary data, not needed in subsequent processing steps, such as dates that tag the sensed data. The resulting physiological data record is stored in a new text file as a comma-separated string of bytes.

3.3 Machine-Learning Approaches for Detection

The previously discussed methods can be used to generate a text file that contains multiple records of physiological data released by the wearable devices to a mobile device. We use this information to build a training data set for training classifiers that map wearable device data to users, their fitness activities, and the output of the mobile device application pertaining to the activity including calorie, steps, or walk count.

The training data set consists of a list of records each stored on a separate line in the file. Each record itself contains the physiological data captured from the wearable device for a performed activity and the labels and values attributed to the user and the recorded activity. The wearable device data are represented as a sequence of comma-separated byte values. To build classifiers that predict the users and their activities using the captured data, we use Weka [6], a publicly available data mining tool. Weka provides a range of supervised machine-learning algorithms for experimentation. It allows users to import training data and run them through specific algorithms. Users can select the input and output fields in the data set on which they want to train the algorithm. We chose to experiment with Wekas implementation of a Decision Tree, Random Forest, Support Vector Machine, and a Naïve Bayesian classifier as these algorithms work best with small training data sets. Once the selected learning algorithm completes the training of a classifier for a single prediction, it can be subsequently used to make predictions about newly captured synchronization records from a wearable device.

4 Experimentation and Results

We have conducted an initial experiment involving three of the five authors from this paper as participants of the experiment. Each participant performed different physical activities while wearing the Misfit Flash, that does not use any encryption on its Bluetooth communication channel with a mobile device. After each participant completed a physical activity, he or she synchronized the wearable device with a smartphone and used the methods described above to intercept and transform the physiological data that was sent to the smartphone. The captured data was annotated with 1) the name of the participant that performed the activity, 2) the terrain where the activity was performed, and 3) the level of physical activity as indicated by the step and calorie count shown on the mobile device.

We have labeled the type of terrain as hilly and outdoors, flat and outdoors, and indoors. Furthermore,

Table 1: *Distribution of Training Data Across the Terrain, the Steps, and Calories.*

	Hilly Outdoor	Flat Outdoor	Flat Indoor
Terrain	3	11	15
	Low	Medium	High
Steps	18	10	1
Calories	10	10	9

Table 2: *10-fold Classification Accuracy for Learning Algorithms*

	DT	RF	SVM	NB
Participant	**93.10%**	**93.10%**	79.31%	89.66%
Terrain	**93.10%**	86.21%	72.41%	79.31%
Steps	62.07%	58.62%	55.17%	**68.97%**
Calories	24.14%	**51.72%**	37.93%	41.38%

we have grouped the step and calorie counts into three different groups for the learning algorithms to make predictions of the level of physical activity and calories and not on specific values. For the step count, we labeled values as low, medium, or high when they range from 0 to 1000, 1000 to 2000, or 2000 and above respectively. For the calorie count, we labeled values as low, medium, or high when they range from 0 to 40, 41 to 60, or 60 and above respectively.

In total, we collected 29 training records from three participants that performed physical activities on running, biking in flat and hilly terrain, and walking indoors and outdoors. Each training record consisted of 800 bytes of physiological data captured from the wearable device and labels for the name of participant, the terrain, and the groups for steps and calories. The different groups and distributions of training data are shown in Table 1 above.

We trained classifiers using four different machine-learning algorithms to make predictions on the participant, terrain, step, and calorie levels. For evaluating the performance of the different machine-learning algorithms, we used a 10-fold cross validation test. Table 2 summaries the results. Values highlighted in bold indicate the best performing algorithm.

As the performance results show, we achieved the highest accuracy for predicting the participant using either Decision Tree (DT) or Random Forest (RF) as our learning algorithms. The Naïve Bayesian (NB) algorithm performed best for predicting the level of steps. Finally, Random Forest (RF) performed best for predicting the level of calories burned by each activity. However, we only achieved an accuracy rate of slightly higher than 50% for making predictions about calorie consumption.

5 Discussion

Results of our initial experiment show that it is possible to predict and track a user and his or her fitness activity based on data intercepted from wearable devices over its Bluetooth communication channel. A closer analysis of the results from the decision tree algorithm shows that the pruned decision trees produced for making predictions are shallow in depth, meaning that few values from the captured data were needed to make the predictions. This is a surprising result as the physiological data record sent from a wearable to its mobile device is fairly large. The variables that played a role in making a decision differed across the different decision trees with the exception of the decision tree for predicting the participant and the terrain, which had a small overlap. This indicates that the byte values in the captured data packets likely represent different sensor values. The reason for the overlap between the decision for predicting participant and terrain may be the lack of variation of activities across participants. Additional training data with a greater degree of variation in the level of activities and the terrain in which the activity was performed will be needed to further assess the predictive quality of the values in the captured data.

Many wearable devices advertise their presence over the Bluetooth channel making themselves visible to potential hackers that can exploit the security weaknesses to capture data from the devices. In this project, we have exploited this weakness and tricked a fitness tracking devices into releasing its data record. With no encryption in place, we were able to capture physiological data recorded by the device and classify it, which constitutes a direct attack on the privacy of the user data. However, by simply not advertising services to nearby Bluetooth devices unless the user specifically requests it, for example when the device must be paired with a mobile device, by requiring authentication when services on the wearable device are contacted, we will make it difficult for hackers finding and controlling the unauthorized release of data of the wearable device. In addition, any randomness added to the format of the data will make it difficult to make sense of intercepted data. Such methods of data randomization can be effective to prevent reverse-engineering of data formats and the infringement of privacy on data sent through the Bluetooth channel.

6 Solution Outline

Below, we outline two simple and energy efficient schemes that can provide reasonable level of security to wearable devices under the constraints of keeping power consumption and time delay of the response time to a minimum. As mentioned above, one of the key reasons we were successful in our attack was a lack of authentication mechanism on the wearable device before it releases the data. For a wearable device energy is at a premium as with any other battery-powered device and hence many sophisticated and proven authentication mechanisms [1], [12] are not suitable while others such as PINs and biometrics are not feasible due to the small form factor. This leads to some manufacturers to completely drop authentication in favor of low latency and energy saving.

We propose a simple authentication mechanism based on One-way hash functions as shown in Algorithm 1 and described below. When a wearable is first synchronized with a smartphone a secure secret (S) is generated on each device. The secure secret can be generated by any key exchange protocol such as the well known Diffie-Hellman protocol. We argue that, initial synchronization of a wearable and smartphone generally takes place in an environment where power is abundantly available and therefore the overhead of set up is not an issue. When a mobile device wants to connect with the wearable it generates a digest D_s

$$D_s = h(S||t_i)$$

and sends it to the wearable. Where $h(.)$ is a one way hash function, S is the shared secret and t_i is the current time epoch. The wearable then generates its own digest D_w using the shared secret it has and the current epoch it knows. If both the digests match, the wearable goes ahead with the connection request, otherwise it drops the request. Since the wearable is time-synchronized with the mobile device with which it was paired, both devices will have the same value of the epoch t_i. Both devices also have the shared secret and therefore the authentication will only work for the wearable and the paired mobile device. This will stop the wearable from responding to requests from arbitrary devices. This scheme requires only two hash operations, one each on the wearable and the mobile device. The execution time of SHA-256 hashing algorithm on Cortex M series processor, which is the processor on Misfit Flash is 1 msec [7]. The authentication scheme therefore only introduces a delay of 2 msec and a corresponding energy consumption of 25.8 μJ on each device. Thus a simple low latency and low energy algorithm can stop random triggering from the wearable.

Algorithm 1 Authentication Algorithm

```
Require: modulus p and $G$ on both devices, time epoch $t_i$ on both devices
  Shared Secret Generation
  The mobile device generates an integer $d$ and sends $X = G^d mod$ p to the wearable
  The wearable generates an integer $e$ and sends $Y = G^e mod$ p to the mobile device
  Both devices calculate the shared secret $S = X^e mod$ p $= Y^d mod$ p
  On mobile device
  Generate digest $D_s = h(S||t_i)$
  Transmit $D_s$ to the wearable
  On Wearable
  Generate digest $D_w = h(S||t_i)$
  if  $D_w == D_s$ then
      Accept the request.
  else
      Drop the request.
  end if
```

Similar to the problems with authentication, some manufacturers forego data encryption due to latency and power constraints. However other more energy efficient randomization methods can be used to provide information theoretic security to the data. Below we outline a simple randomization scheme based on CMT encryption [3] which is semantically secure. The CMT scheme encrypts data as

$$Enc_k(d) = (d + f_k(r)) mod \text{ M}$$

Where d is the data being encrypted, $f_k(.)$ is a pseudo random function (PRF) based on key k which uses the nonce r to generate a random number and M is an integer which is a system parameter. CMT encryption basically adds random noise to the data mod M, and is secure under the assumption that $f_k(r)$ is random and not disclosed to any third party. We can substitute$f_k(r)$ with digest D_w on the wearable to add random noise to the data. Digest D_w uses the shared secret S as the key and the use of the time epoch t_i means the value of D_w will change with every epoch. This random noise in the form of the digest is secure as long as the one way hash function $h(.)$ is secure. Thus our randomization function on the wearable will then be,

$$d_r = (d + D_w) mod \text{ M}$$

where d_r is the randomized data. This randomized data will then be transmitted to the mobile device once the wearable is successfully triggered after authentication. Once the mobile device receives the randomized data, it can easily remove the random noise since it knows the shared secret S and the time epoch t_i by calculating,

$$d = (d_r - D_s) mod \text{ M}$$

This randomization method also involves only two hash operations on each side thus introducing only a small amount of latency and energy overhead compared to traditional symmetric and asymmetric key encryption techniques.

7 Conclusion and Future Work

Wearable devices are widely used to collect health-related information of its users. Unfortunately, the devices often lack proper implementation of security protocols to protect the privacy and integrity of the data as the data are exchanged with other devices over a wireless network. In this paper, we have shown how a wearable device, specifically the Misfit Flash, that does not use encryption or authentication to synchronizing data with its mobile device can be tricked into releasing its data on demand by a nearby hacker. We have further shown how the data can be collected and analyzed using machine learning techniques to identify individuals and track their behavior.

The next step in our project will be to include a larger pool of participants and to vary the fitness activities to generate a larger training data set for experimentation. Furthermore, we will experiment with applying and adapting the presented methods to other, widely used wearable devices to study their vulnerabilities and evaluate our proposed solutions in improving the security of the devices.

References

[1] S. M. Bellovin and M. Merritt. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 72–84, May 1992.

[2] R. Bouhenguel, I. Mahgoub, and M. Ilyas. Bluetooth security in wearable computing applications. In *2008 International Symposium on High Capacity Optical Networks and Enabling Technologies, Penang*, pages 182–186, 2008.

[3] Claude Castelluccia, Aldar C-F. Chan, Einar Mykletun, and Gene Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(3):20:1–20:36, June 2009.

[4] Britt Cyr, Webb Horn, Daniela Miao, and Michael Specter. Security analysis of wearable fitness devices (fitbit). *Massachusets Institute of Technology*, page 1, 2014.

[5] Y. L. Zheng et al. Unobtrusive sensing and wearable devices for health informatics. *IEEE Transactions on Biomedical Engineering*, 61:1538–1554, 2014.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[7] Safwat Mostafa Noor and Eugene John. Performance and energy evaluation of arm cortex variants for smart cardiac pacemaker application. In *In Proceedings of the 2016 Int'l Conf. Biomedical Engineering and Sciences, BIOENG'16 —*, pages 13–18, 2016.

[8] A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2010.

[9] G. Press. Internet of things (iot) predictions predictions from forrester-machina-research wef, gartner idc. *Forbes Magazine*, January 2016.

[10] Thomas Reichherzer, Amitabh Mishra, Ezhil Kalaimannan, and Norman Wilde. A case study on the trade-offs between security, scalability, and efficiency in smart home sensor networks. In *Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, page in press, 2016.

[11] M. Tan and K. A. Masagca. An investigation of bluetooth security threats. In *2011 International Conference on Information Science and Applications*, pages 1–7, April 2011.

[12] Thomas Wu. The secure remote password protocol. In *In Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium*, pages 97–111, 1998.