NETWORK SECURITY PROJECT

EXPOSING ENCRYPTED WIRELESS DATA TRANSFER IN WEARABLE IOT DEVICES

SMARTWATCHES AND FITNESS BANDS

SUBMITTED BY:

SUMISH PAL SINGH AJMANI (ssa8737)
KARAN PARIKH (kap9580)
CHINMAY TOMPE (cst9314)
SAHIL CHITNIS (ssc9983)

BACKGROUND

Very few technologies exist today that have redefined personal fitness as we know it.

Smartwatches and fitness bands have become a staple in our daily life and we use them as if they are an extension of us. They remain on the wrist, calculating personal and health data every few seconds to help us better understand our body and reach our fitness goals.

Many widely used devices such as the Apple Watch or the Xiaomi / Lenovo watches (famous in the android market) have native apps running on them or have companion smartphone apps. **Most of these apps do not perform TLS certificate validation.**

This can grant an unauthorized user (an attacker) access to critical information such as exercise information, heart rate, body data and sometimes even financial information such as debit card details, balance etc. The attacker can use this information for unimaginable malicious activities.

Sometimes the apps running on these watches **do not perform basic data integrity check techniques** such as hashing which can give out user credentials out in plaintext. These wearables connect to smartphones via bluetooth, which is another avenue for sniffing and potential data leak.

Such a design that favors openness in security can lead to unrepairable damage to the users as well as the manufacturers reputation in the market.

WHAT DID WE DO?

Since different manufacturers have different ways to design the software of the wearables and the applications that go along with it, we tested 4 devices by performing **MITM attacks as well as Bluetooth Sniffing** on them to check the security of its wireless connection.

We decided to do this in 2 phases:

Phase 1 (MITM Proxy):

- Intercept traffic between smartwatch and server using MITM proxy.
- Intercept traffic between the smartwatch app and the server using MITM proxy.
- Try to devise methods to do this in the real world.

Phase 1 includes setting up an **LT2P** with **IPSEC VPN** server on the Cloud VM and connecting the wearable and/or smartphone to the VPN server. For all the 4 devices, we installed the MITM proxy certificate on the iPhone to test each device's companion apps. (Known issue with Android : https://github.com/mitmproxy/mitmproxy/issues/2054)

In terms of directly installing the **MITM proxy certificate on the wearable**, we were only able to install it on the Apple Watch. Phase 1 tools include- MITM proxy, IPSec VPN, VirtualBox.

Phase 2 (Bluetooth Sniffing):

- Intercept Bluetooth data traffic between watch and phone.
- Explore Bluetooth 4.0 classic relay attack with Raspberry Pi & Gattacker

Phase 2 involves generating **"bug reports"** for each device using **Android Debug Bridge** command line tools when connected to an Android smartphone.

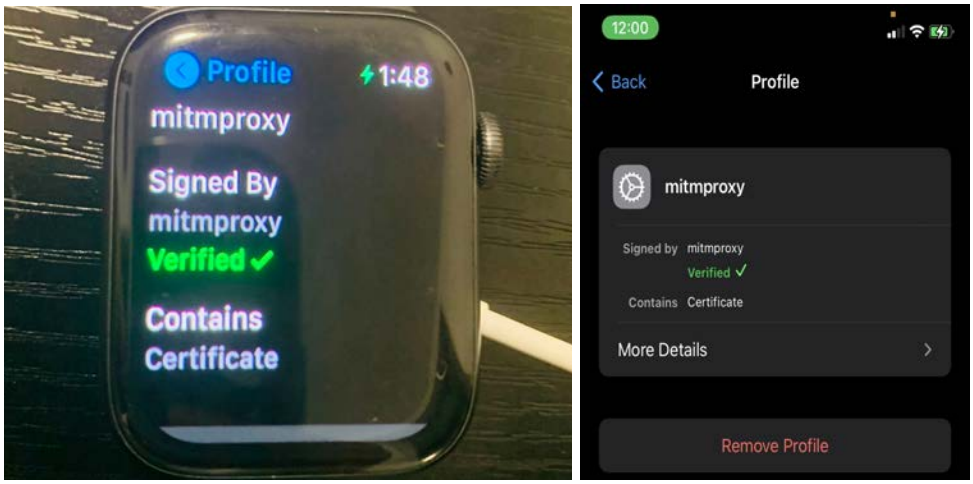
These bugreports contain **"bluetooth logs"** which can then be opened up in **Wireshark** to study Bluetooth packets between the smartphone and the wearable. Phase 2 tools include Bluetooth, Android, Wireshark.

WHAT DID WE DISCOVER?

We are listing down some of the most sensitive data that we could see on these devices while performing MITM proxy and Bluetooth Sniffing.

DEVICE 1: APPLE WATCH SERIES 6

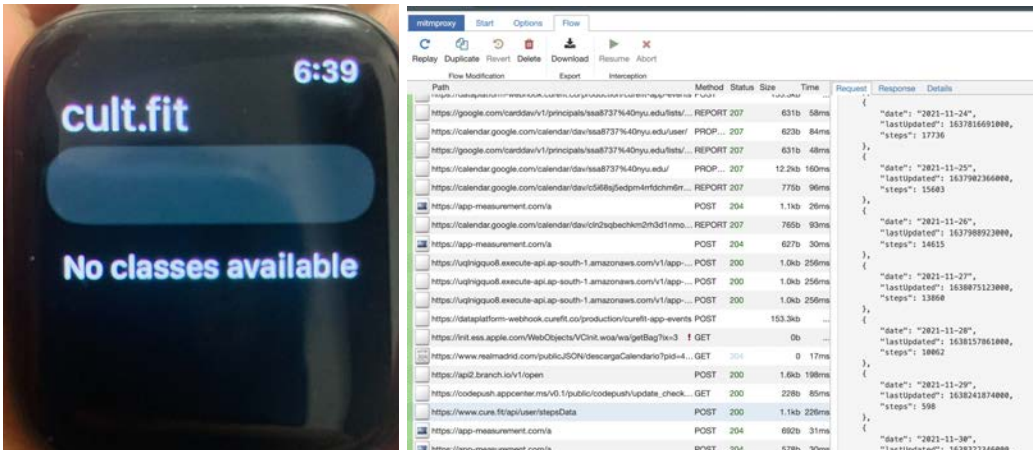
a. Installing MITM proxy certificate



MITM Proxy Certificate is successfully installed on both the Apple Watch & the iPhone

b. CultFit Fitness App

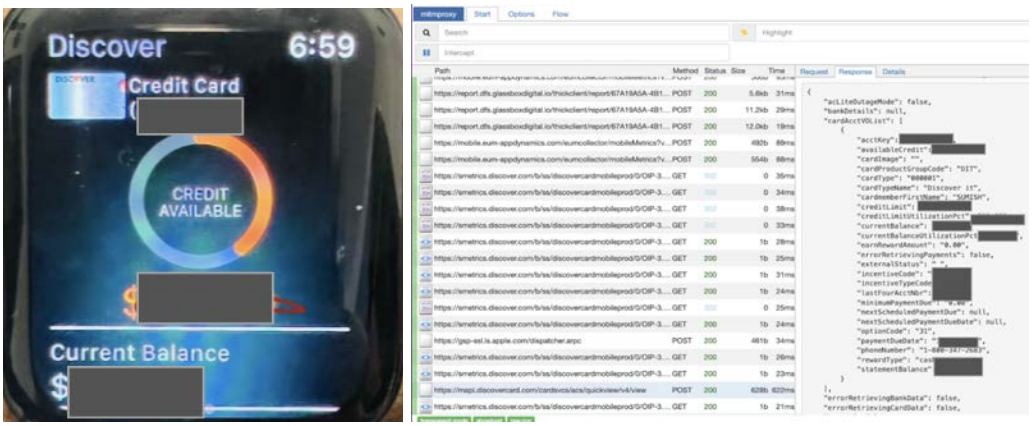
This is a fitness app in India where you can schedule your online classes, track your steps, daily health, heart rate, and personal hygiene.



MITMweb interface shows that traffic is intercepted and was able to extract the number of steps taken by the user

c. Discover Card App

This is an app of a credit card company which allows users to keep track of their transactions, make payments using the watch, check credit history and other user sensitive information.

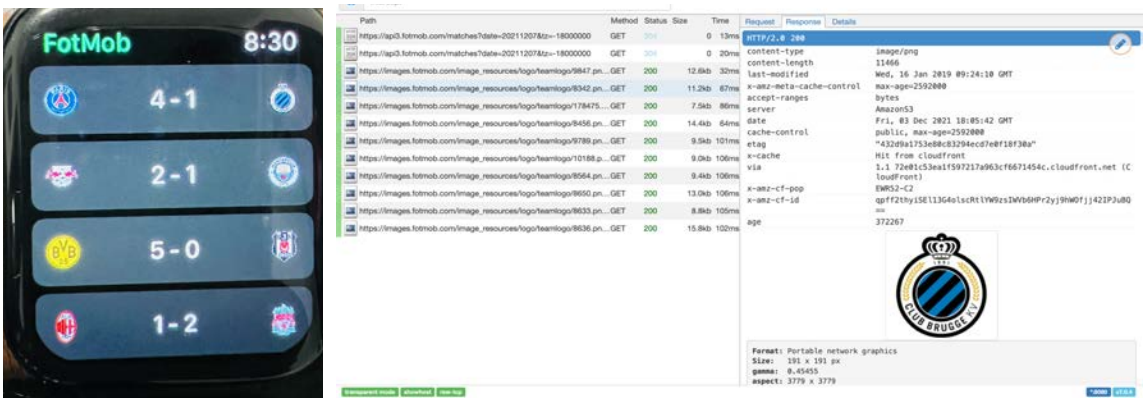


MITMweb interface shows credit card details and other user sensitive information in plain text

d. FotMob App

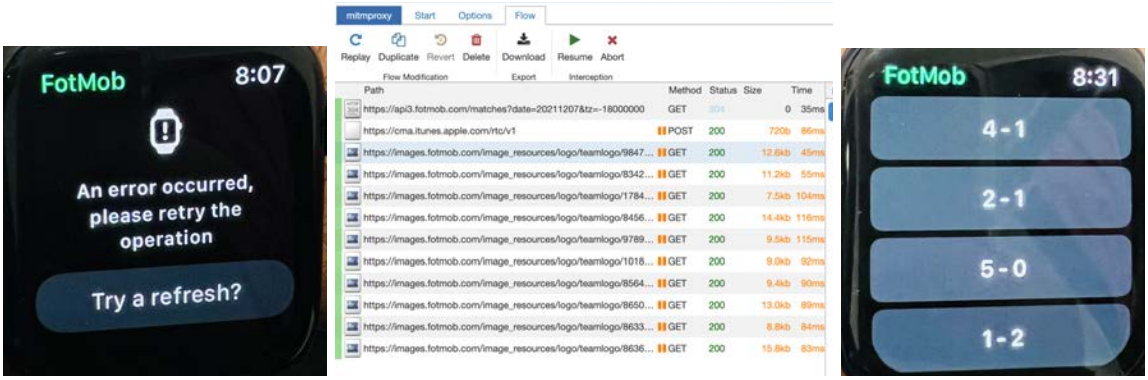
This is a football news app which shows live scores and latest football news. Things turnaround during our analysis when we were able to *not only intercept but also alter the displayed data* on the fly. Here are the stepwise observations of the MITM attack performed on the FotMob App.

➔ Observation-1 (Before MITM Interception)



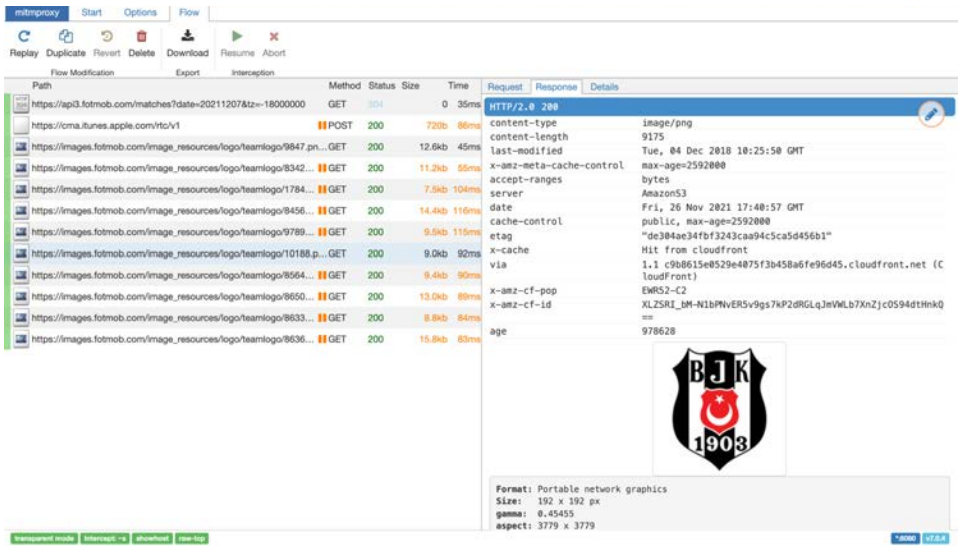
Football scoreline data, team logos of the live matches are displayed as expected, status is Green and 200 OK

➔ Observation-2 (During MITM Interception)

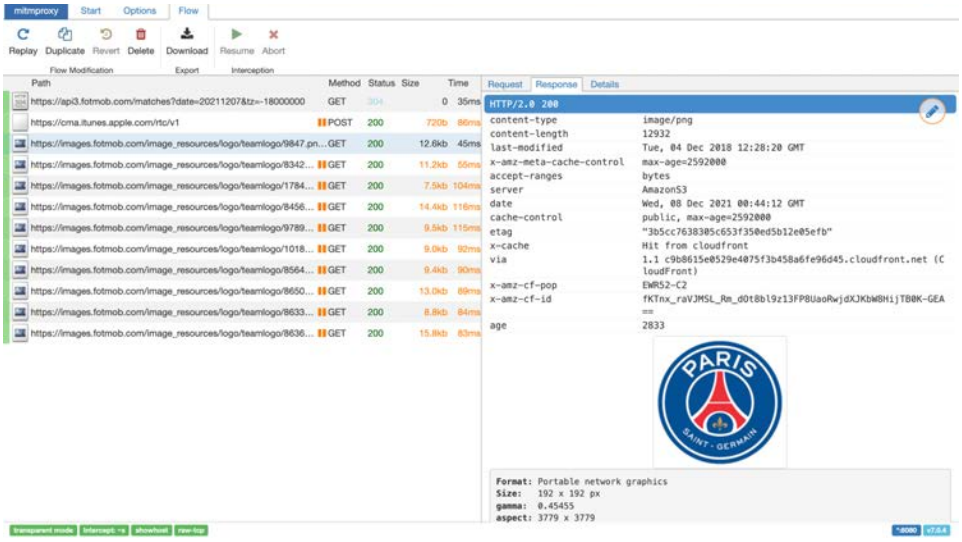


Network is intercepted, app is not able to load the data temporarily, status paused during HTTPS connection

➔ Observation-3 (After MITM Intercepted)



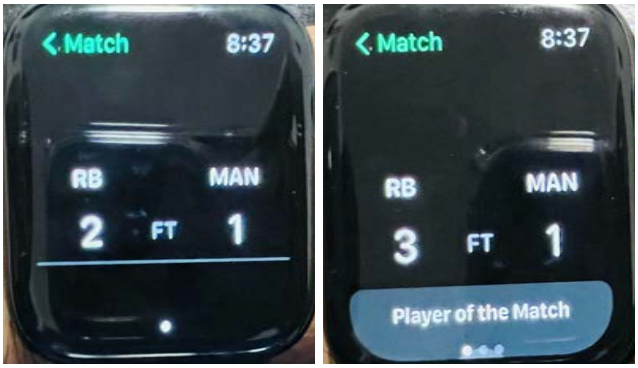
Randomly selected packets have been released and status is Green



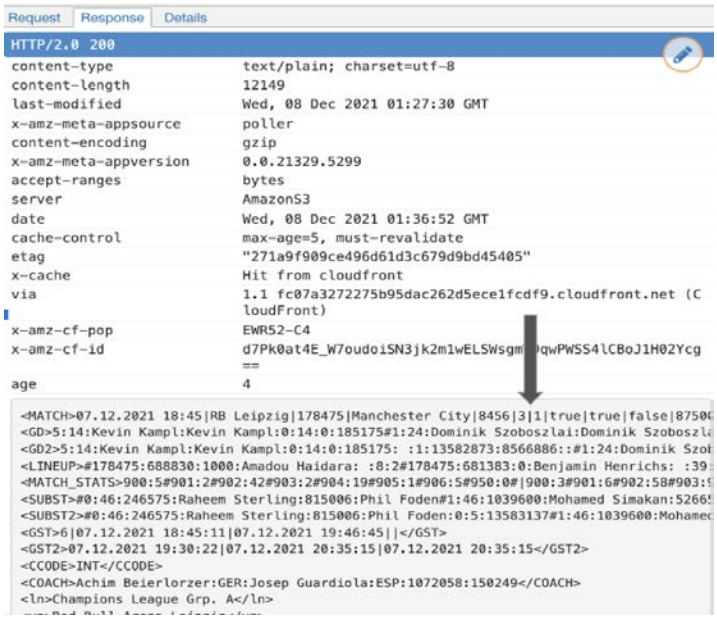
Randomly selected packets have been released and status is Green

Team logos and scores are now visible for randomly selected packets

➔ **Observation-4 (After MITM Alteration)**



Real time scoreline of RB Leipzig vs Manchester City (2-1) altered to (3-1) on the fly



Packet alteration using edit option in MITMweb interface

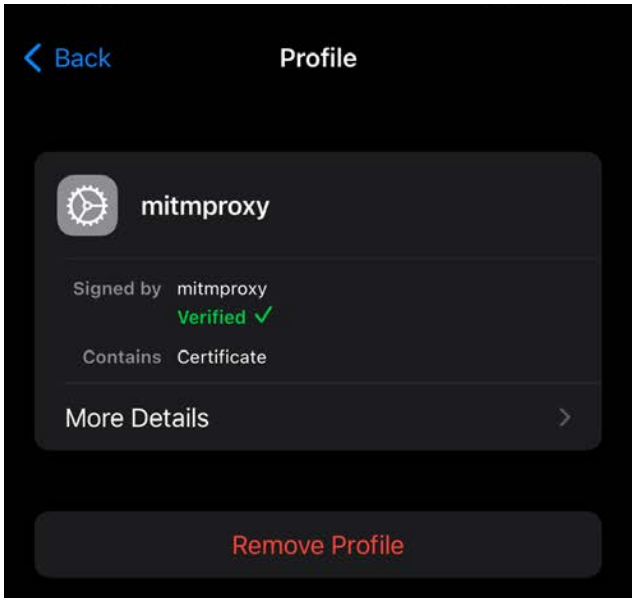
e. Bluetooth Sniffing

➔ Apple Watches can only connect to iPhones.
➔ Open source Bluetooth sniffing tools are unsupported or unavailable in OSX
➔ **Enhancements?** Use Apple Developer Tools such as XCode (paid services) to get access to various tools used for Bluetooth sniffing.

Tools like BetterCap, Bluez didn't displayed any devices in MacOSX

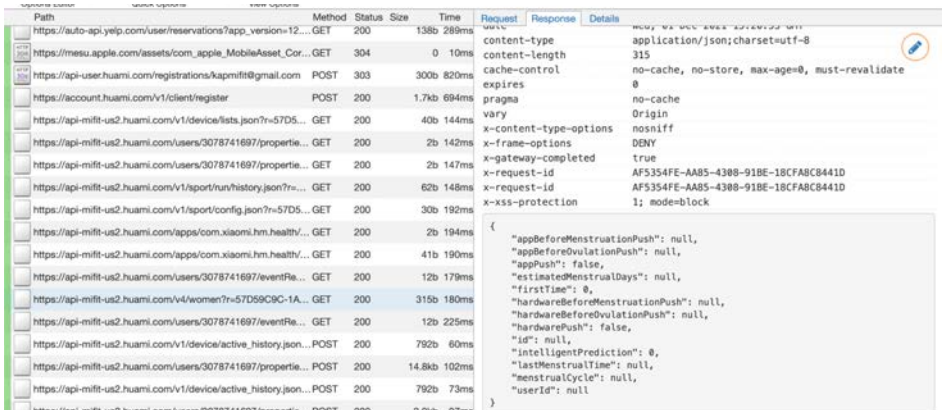
DEVICE 2: XIAOMI SMART BAND 5

a. Installing MITM proxy certificate

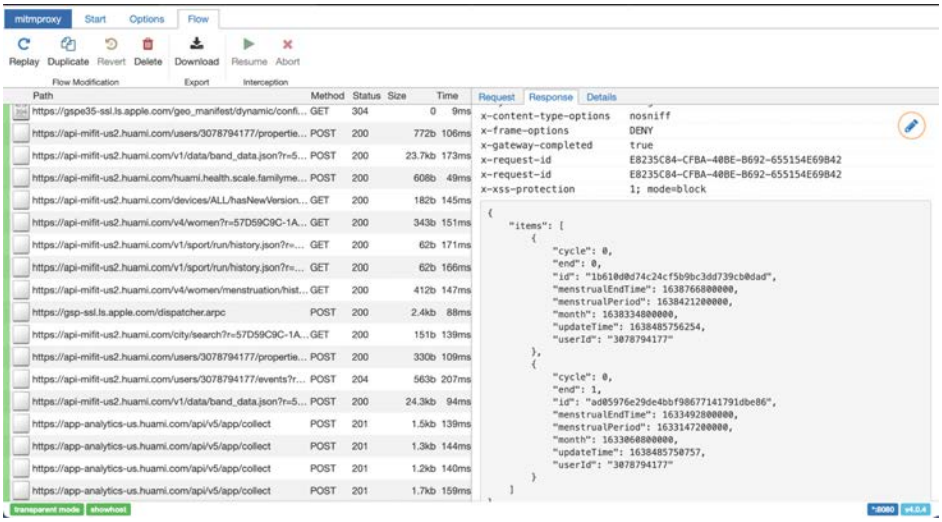


MITM Proxy Certificate is only installed on the iPhone as band cannot connect to the internet at all

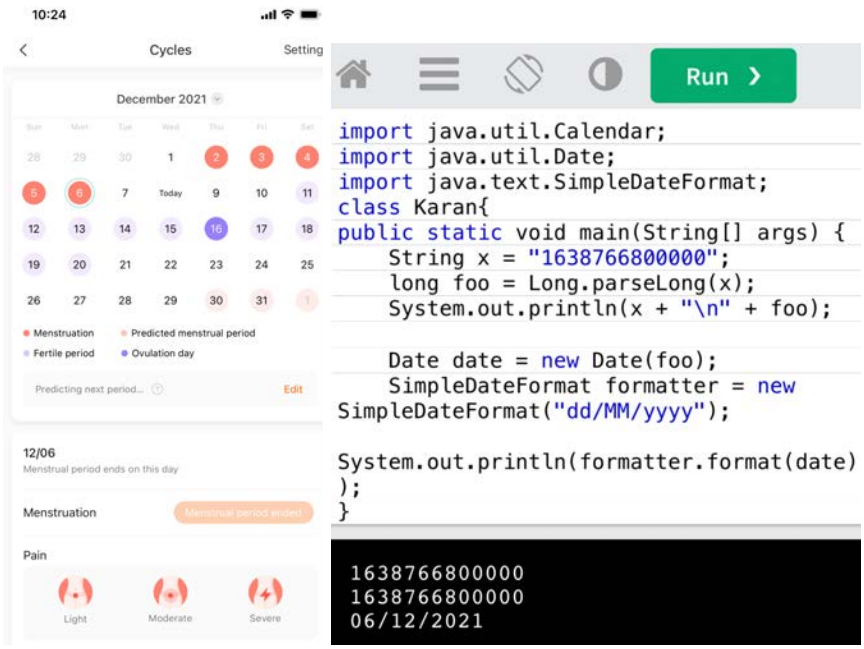
b. Female Menstrual Data exposed (MITM Proxy)



Female profiles on the MiFit app store menstrual data which is seen in plain text

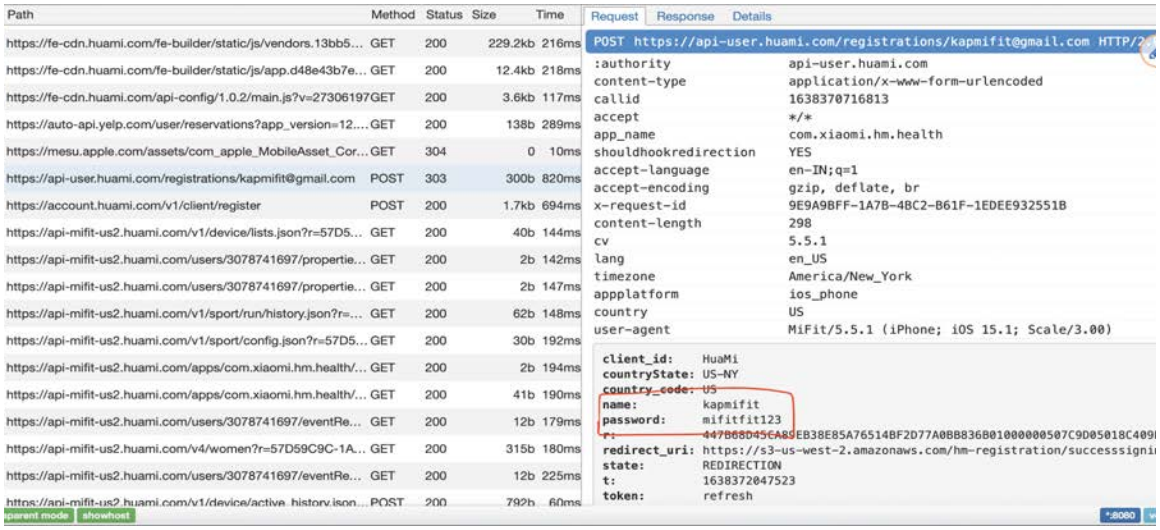


Every time the MiFit app is loaded, menstrual data is sent to the server which includes: Menstrual month, cycle end date, menstrual period, user ID, etc..



Menstrual data as seen in MiFit App (L.) Date exposed in long data type can be easily converted to date format (R.)

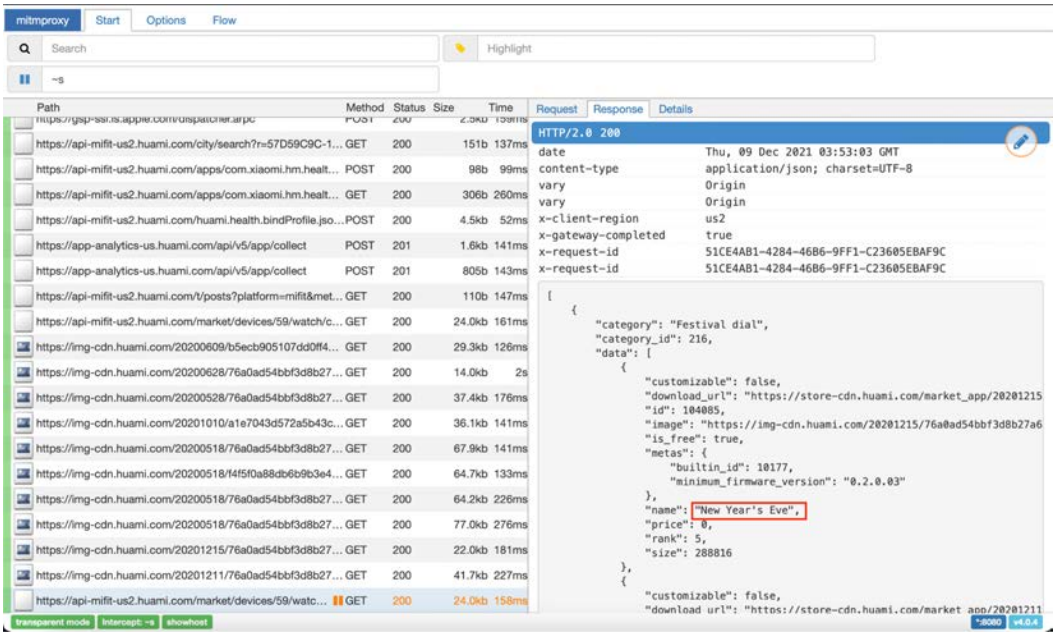
c. User account details for MiFit app exposed (MITM Proxy)



Login username and password for MiFit account sent in plaintext

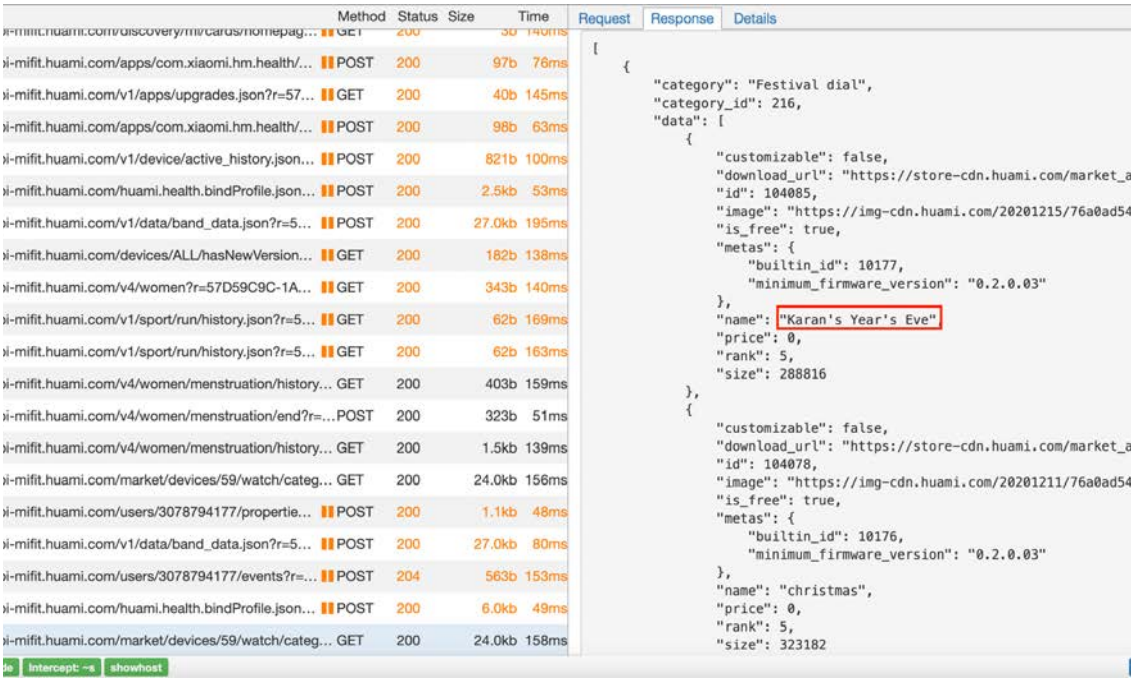
d. Watchface store on the app can be altered (MITM Proxy)

➔ Observation 1: Intercepting WatchFace Store response packet



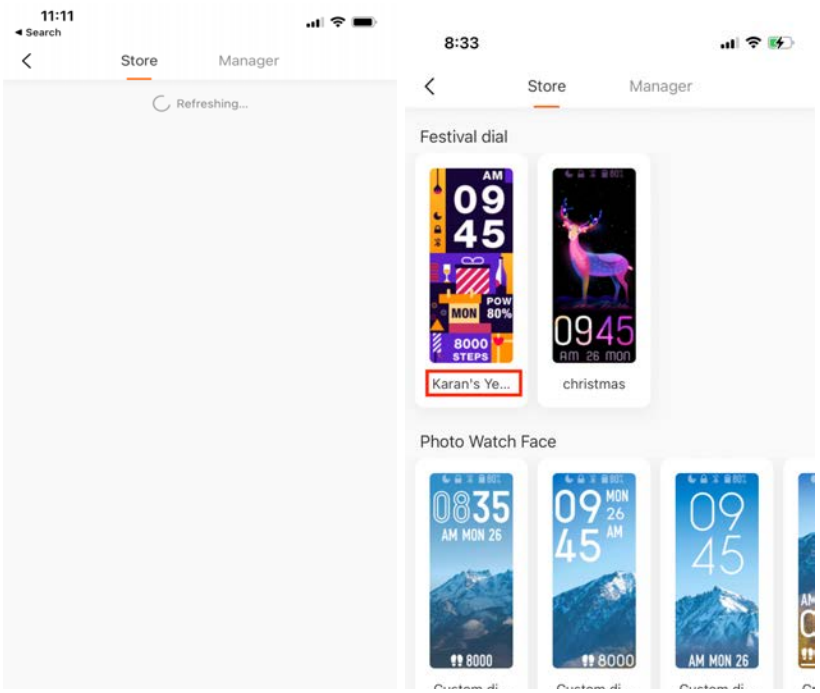
WatchFace Store on MiFit App is loaded in Javascript as seen above. Value of "name" will be altered on interception

➔ Observation 2: Altering the response packet for Watchface Store



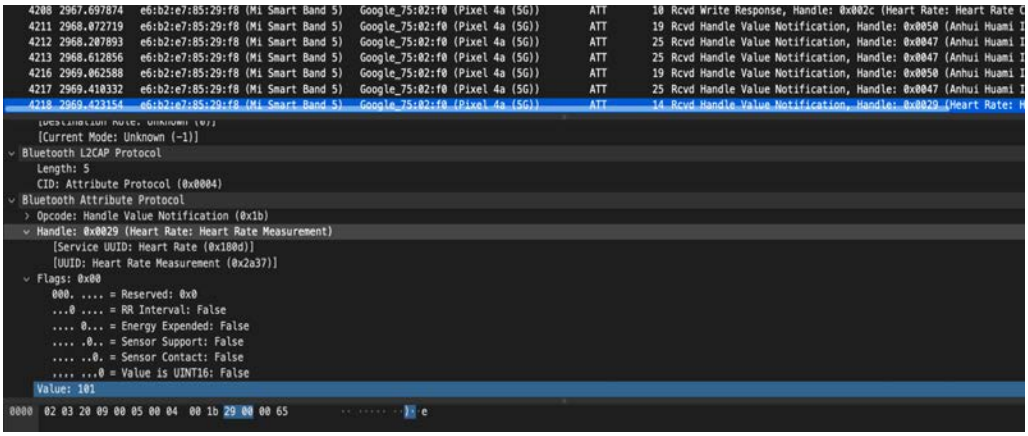
Value of "name" is altered on the fly from "New Year's Eve" to "Karan's Year's Eve" and then the packet is allowed to be sent to the MiFit App

➔ **Observation 3: MiFit App as seen during packet alterations**

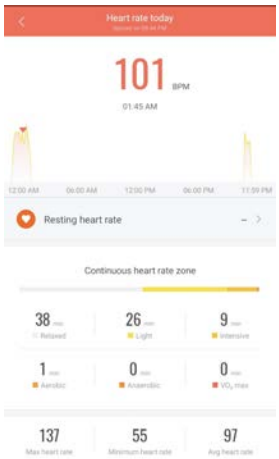


Watchface Store on MiFit App during interception of response packet(L.) Watchface Store after response packet alteration (R.)

e. Heart rate data sent to MiFit App is seen during Bluetooth Sniffing



Heart Rate data seen in bluetooth packets when data is synced between the band and the MiFit app (value=101)

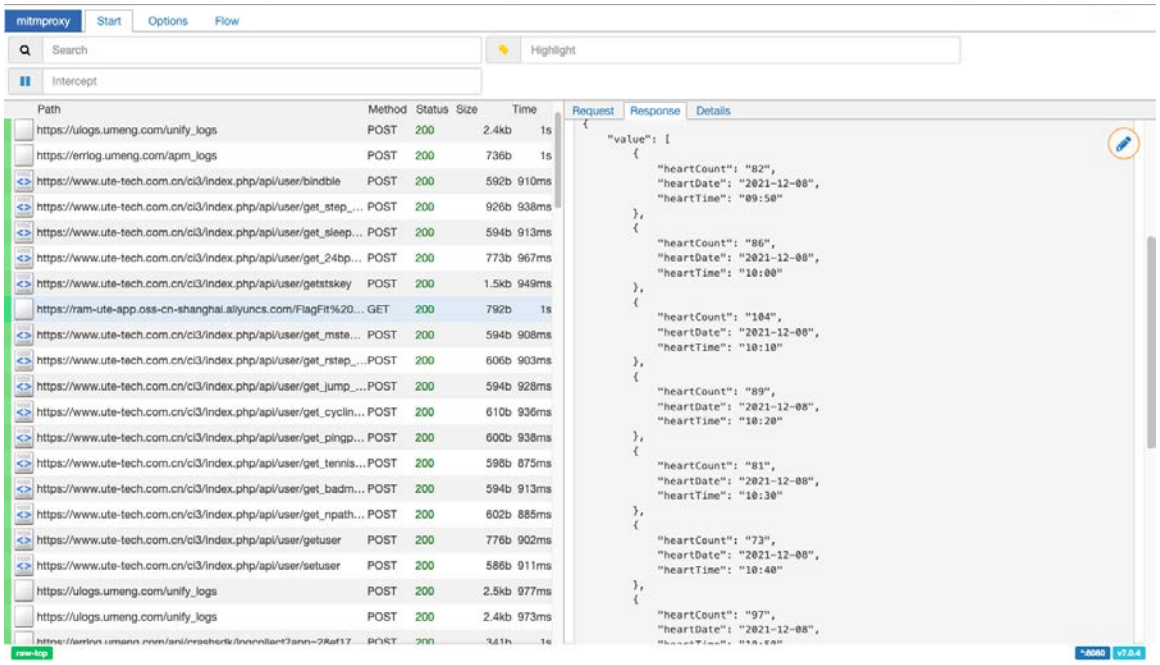


Same heart rate data seen in MiFit app at the time of syncing

DEVICE 3: L8STAR SMARTWATCH

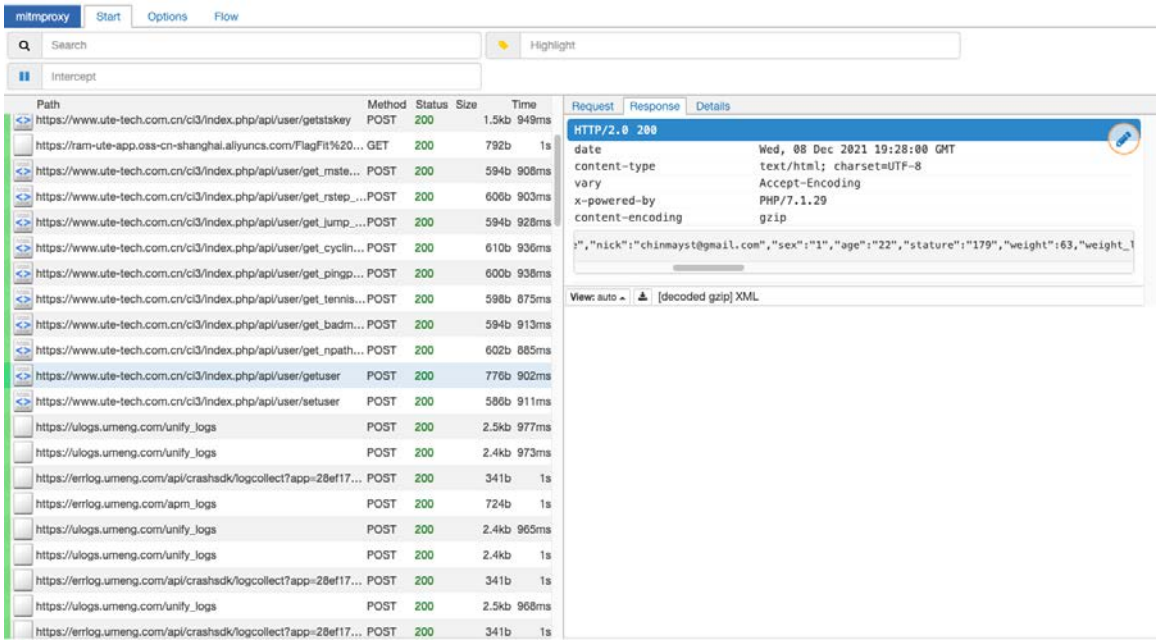
The L8Star watch does not connect to the Internet itself. MITM proxy was thus only installed on an iPhone to study packets sent by its companion app (L8Star).

a. Heart Rate Count, Date and Time is exposed (MITM Proxy)



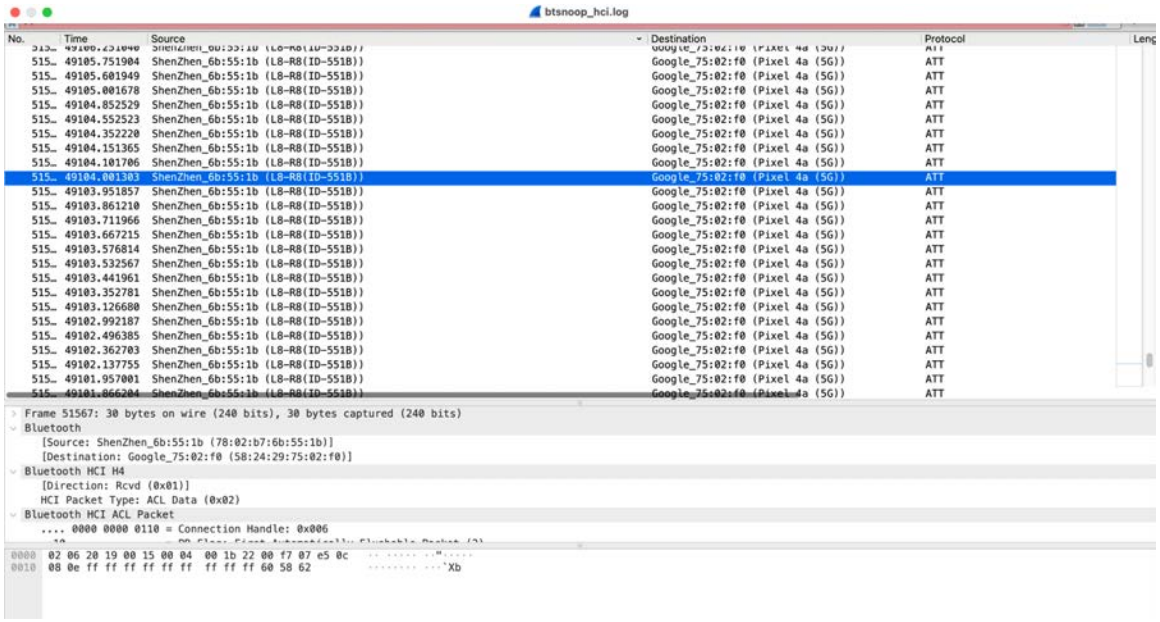
Heart rate data, date and time of measurement is seen in plaintext when the app communicates to the server

b. User sensitive data such as email ID and personal details are exposed



Personal details of the user exposed

c. Bluetooth sniffing results

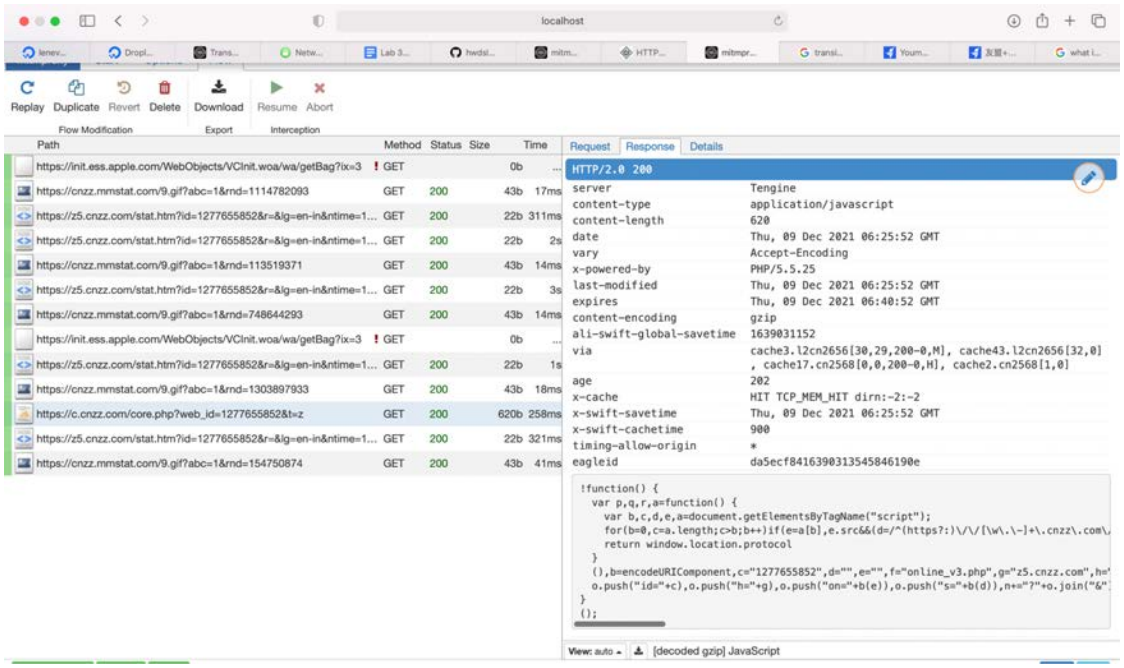


Bluetooth: No plain text observed, everything is indecipherable.

DEVICE 4: LENOVO WATCH S2

The Lenovo Watch S2 does not connect to the Internet itself. MITM proxy was installed on an iPhone. The Lenovo app (LenovoSmartWatch) sends almost no local data to the cloud. No email verification during signup as well. Bluetooth Sniffing provides data that is indecipherable.

a. Lenovo Watch app sends data to CNZZ.com (MITM proxy)



cnzz.com is an Ad and Marketing analysis company

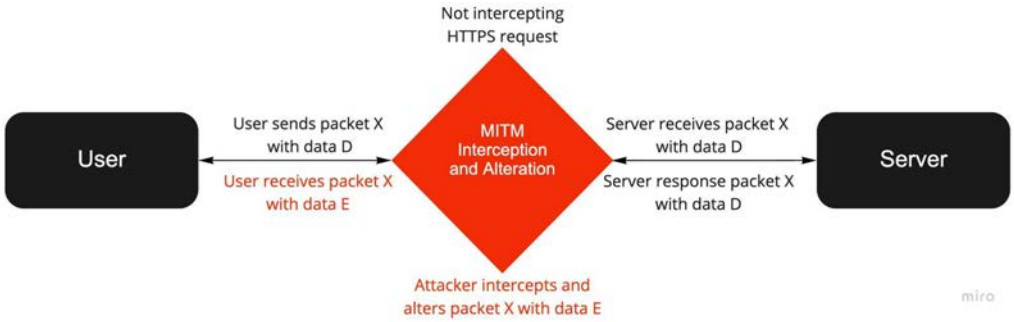
KEY TAKEAWAYS

- **Apps should add certificate validation steps to protect user privacy**
  - The installation of the MITM certificate to the user's phone or IoT device was the key step in our attack.
  - Despite adding MITM certificate to the list of root certificates, only a few apps and device features were inaccessible, meaning those apps or features are validating the authentication of root certificates or using the certificate pinning technique.
  - For the case of WatchOS, **7 out of 10** third party apps aren't validating certificates, leading it to the disclosure and alteration of sensitive data. The rest of the 3 apps were associated with Apple Inc and linked to iCloud servers, performed some validation which led to the breaking of the TLS handshake.
  - For the rest of the fitness bands, in total **15 out of 20** features were exposed simply because corresponding watch apps aren't validating certificates.

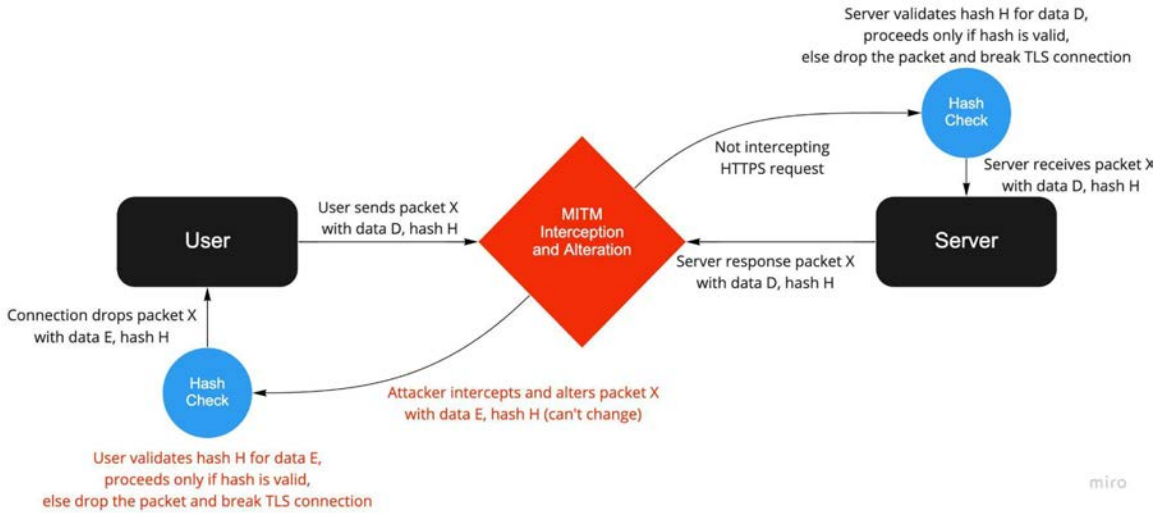
- **Bluetooth pairing, exchange of data, and access controls shall be encrypted**
  - The mere access to the free Android Developer Bridge (ADB) tool in Android devices allows bluetooth sniffing between an Android smartphone and the wearable.
  - Apple watches only connect to iPhones, so it was not possible to perform any sort of analysis. Though we are hopeful that paid access to the iOS developer tool (XCode) might lead us to interception of bluetooth connection.
  - For the case of Xiaomi Smart Band 5, we observed the exchange of all data in plain texts. Moreover, we were able to **see actual write commands** over bluetooth connection to the fitness bands leading to serious privacy threat over user's sensitive data.
  - For the cases of Lenovo and L8Star fitness bands, the exchange of data was indecipherable.

- **Include Data Integrity checks inside packets having user sensitive data**
  - **Idea:** Including data integrity checks such as hash values of sensitive data inside the packet might solve the problem of interception and alteration of HTTPS packets even if certificate validation is not being performed.

- **Working**: The following flow charts showcase scenarios before adding and after adding integrity checks:



Successful attack without including hash values in a packet



Attack unsuccessful due to inclusion of hash checks, drops the packet

- **Our Opinion:** Though the implementation of inclusion of hash data and it's validity check on each packet is a big task, it will significantly decrease the performance of MITM attacks even after certificate installation. This mitigation has trade off with time and space complexity over internet connection between user and the app's server, but an outstanding improvement of data security.

SUMMARY TABLE

Were we able to…	Apple Watch	Xiaomi Smart Band 5	Lenovo S2	L8star Band
Install a MITM certificate on watch and make it trusted?	Yes	No. Incompatible	No. Incompatible	No. Incompatible
Expose Fitness app user credentials ?	Yes. Cultfit app	Yes MiFit App	No. Data stored locally	Yes
Expose user personalized Health Data (Heart rate, Menstrual data)?	Yes. Heart rate, step count	Yes. Heart Rate, Menstrual Data	No. Data stored locally	Yes. Heart Rate, step count
Expose Financial information ?	Yes. Credit card info & Balance	Not supported	Not supported	Not supported
Expose the current location ?	Yes. Uber location	Yes. Weather Location	No. Data stored locally.	No. Data stored locally.
Alter Sensitive Information (Violate Integrity)	Yes. Change live Football scores	Yes. WatchFace Store in-app altered	Not Supported	Not Supported
Expose data shared via bluetooth ?	No. Require additional resources	Yes. Heart Rate, Weather Data.	Yes. Watch Basic Info, mostly indecipherable	Yes. Watch Basic Info, mostly indecipherable
Intercept data via a self signed trusted certificate installation and get system information ?	Yes. Certificate installed on iPhone.	Yes. Certificate installed on iPhone.	Yes. Certificate installed on Android phone.	Yes. Certificate installed on Android phone.
Ease of performing attack / Scalability	Easy but currently requires access of physical devices to install certificates or obtain bluetooth log			
Expose data communicated to Marketing & Advert firms ?	Yes.	Yes. Facebook Ads	Yes. To chinese companies	Yes. Facebook Ads
Expose data for other apps installed on watch ?	Yes. Apple store, Uber, FotMob, Discover, Spotify	No. Incompatible.	No. Incompatible	No. Incompatible

FUTURE ENHANCEMENTS

- **Open-WiFi MITM Proxy attacks:**

Our next logical step to advance our MITM proxy attack would be to try to do it without needing physical access to a victim's smartphone (to install the MITM proxy certificate). A way to do this could be using a command line tool called "berate_op" to create an open Wi-Fi network by configuring an antenna connected to our machine. Then, we could use captive portal technology along with social engineering to lure the victim to install the MITM proxy certificate in exchange for promising connection to our free open Wi-Fi network. This would be a simulation of a possible real world attack.

- **Gattacker: Node.js package for BLE security assessment (Bluetooth MITM)**

Creates an attacker interface for exploiting packet exchanges during bluetooth connection by letting the user's device believe it’s connected to the watch and vice versa, but in reality the connection between both of them passes through an intruder device, causing bluetooth packets to get intercepted. This could be a simulation of a possible real world attack.

- **Xcode: Apple iOS Developer Tools**

Similar to Android Debug Bridge tool, one can explore the possibility of intercepting bluetooth connection and exchange of packets between Apple Watch and iPhone using paid services like Xcode.

- **Bettercap: Tool for 802.11, BLE, IPv4 and IPv6 networks reconnaissance**

A better version of Gattacker, using a similar concept but extending it to WiFi and the internet. It provides the following features to the attacker: 1. WiFi networks scanning, deauthentication attack, clientless PMKID association attack and automatic WPA/WPA2 client handshakes capture. 2. Bluetooth Low Energy devices scanning, characteristics enumeration, reading and writing. 3. Passive and active IP network hosts probing. 4. ARP, DNS, and DHCPv6 spoofers for MITM attacks on IPv4 & IPv6 networks.

IMPACTS AND REPERCUSSIONS

Following table describes the impacts of MITM attacks and bluetooth sniffing on smart watches and fitness bands in terms of number of users, materiality, and data protection.

Watches, Fitness bands	Apple Watch	Xiaomi Smart Band 5	Lenovo S2 smartwatch	L8star R8
Number of users	>100M	>100M	>100K	>1M
Materiality (USD)	$40B	$2.2B	$2.4B	$24M
Data Protection (MITM)	No (3rd party apps), Yes (Apple Inc. apps)	No	No	No
Data Protection (Bluetooth)	NA	No (Plain packets)	Yes (Encrypted packets)	Yes (Encrypted packets)

Impacts based on number of users, price of a device in 2021 w.r.t our analysis

IS IT SAFE TO USE THESE DEVICES?

YES!

THE INSTALLATION OF A MITM PROXY CERTIFICATE ON A VICTIM'S SMARTPHONE IS INCREDIBLY DIFFICULT TO DO SINCE AN ATTACKER WOULD NEED PHYSICAL ACCESS TO THE DEVICE TO BE ABLE TO DO IT. EVEN THOUGH SOCIAL ENGINEERING CAN BE EMPLOYED TO LURE THE VICTIM TO INSTALL THE CERTIFICATE, IT IS STILL A VERY DIFFICULT SCENARIO TO ACHIEVE IN THE REAL WORLD.

References

- **https://www.thrivewearables.com/how-5g-will-transform-wearable-technology-and-make-us-healthier/**
- **https://github.com/hwdsl2/setup-ipsec-vpn**
- **https://docs.mitmproxy.org/stable/concepts-filters/**
- **https://github.com/conorpp/btproxy**
- **https://www.conorpp.com/proxying-bluetooth-devices-for-security-analysis-using-btproxy/**
- **https://medium.datadriveninvestor.com/hacking-bluetooth-low-energy-ble-smart-devices-bd58bf56268b**
- **https://www.aboveavalon.com/notes/2021/2/11/apple-watch-is-now-worn-on-100-million-wrists**
- **Image and other references:**
  https://www.wireshark.org/
  https://mitmproxy.org/
  https://www.virtualbox.org/
  https://www.apple.com/watch