

How to shield an IoT product from the OWASP IoT Top 10

or

The "S " in IoT stands for security

OWASP Meetup Cologne, 2023-10-26



Pablo Endres

*Managing Director
Lead Security Consultant*

✉ epablo@sevenshift.de

🐦 @epablosensei

🌐 <https://www.linkedin.com/in/pabloendres>

Experienced security consultant,
Professional Hacker and
Trainer

- Professional Hacker and Security Trainer
 - IoT Security {Bootcamp, Strategy} ICS or IIoT
- Penetration and security testing (design, planning and execution)
 - IoT, IIoT, ICS, Infrastructure, Cloud, Web, Mobile ...
- Security consulting: architecture, secure-by-design, programs..
- Project management
- Certified: CISSP, OPSA, OPST
- Special interest: Karate and slacklining

Agenda

01 - Introduction to IoT

What is IoT?, Architecture

03 – OWASP IoT Top 10

Project, Top 10 list

02 - IoT Security

Current status, attack surface

04 – Use cases

GPS Tracker, Smart Cities



IoT

What is IoT

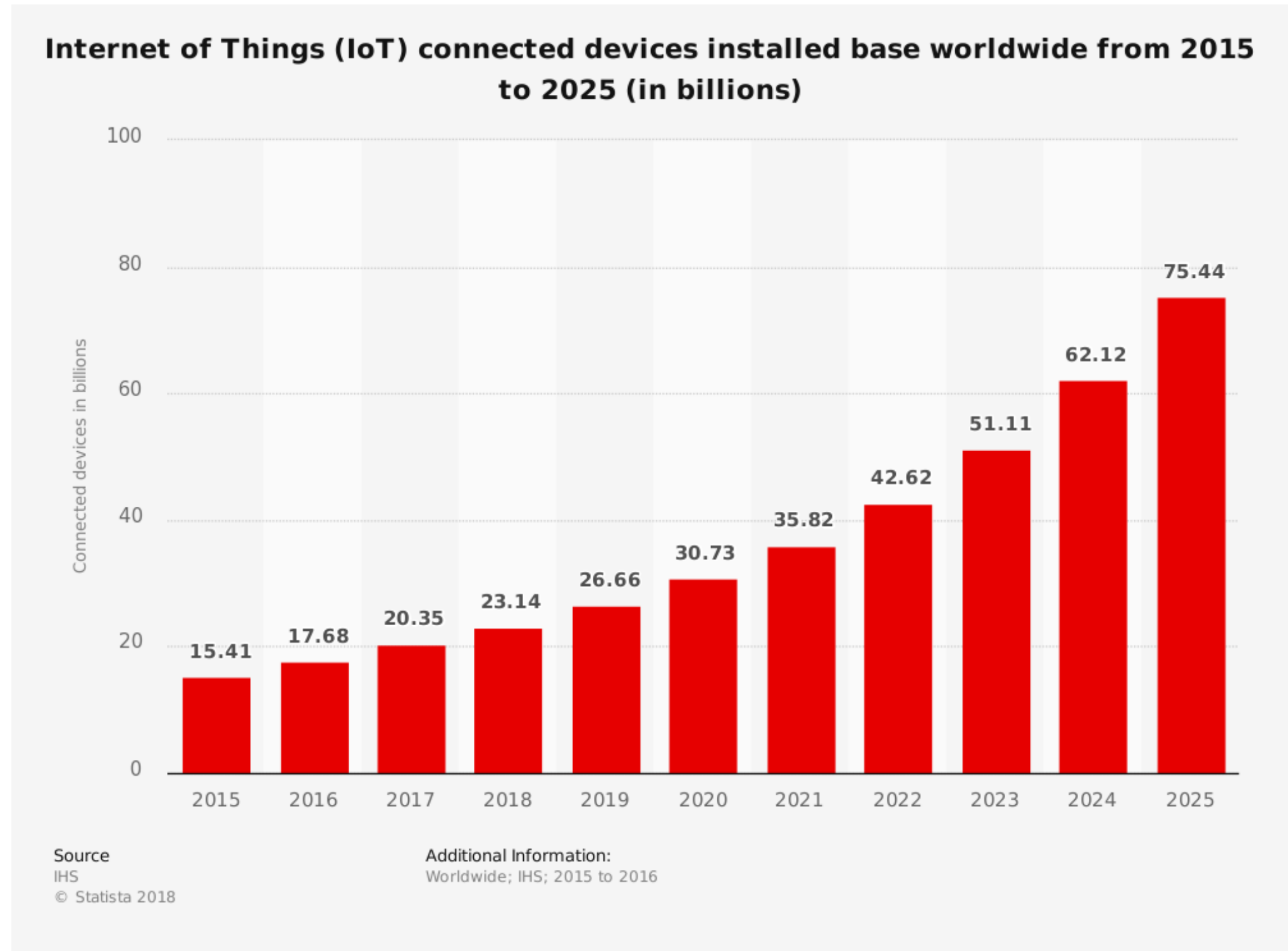
- Internet of Things
- Connected objects interacting with the physical world (sensors / actors)
- Used for automation, monitoring, and data collection purposes
- Consumer IoT: smart watches, plugs, home, etc
- Automotive
- IIoT: smart city (parking), ICS, SCADA, smart grid



https://en.wikipedia.org/wiki/Smart_meter#/media/File:Intelligenter_zaebler-_Smart_meter.jpg

... in times of IoT

- IoT is a big trend, more devices connected every day
- Everything is going online, even what shouldn't



IoT is complex

- a. Devices / Sensors / Things
- b. Communication protocols
- c. Gateways (optional)
- d. Networking
- e. Data collection
- f. Visualization / Action / Applications



IoT is complex (different perspective)

- a. Device hardware (SoC, MCU, sensors, actors)
- b. Device firmware
- c. Connectivity
- d. Mobile applications
- e. Web applications
- f. APIs and backends

We must deal with ..

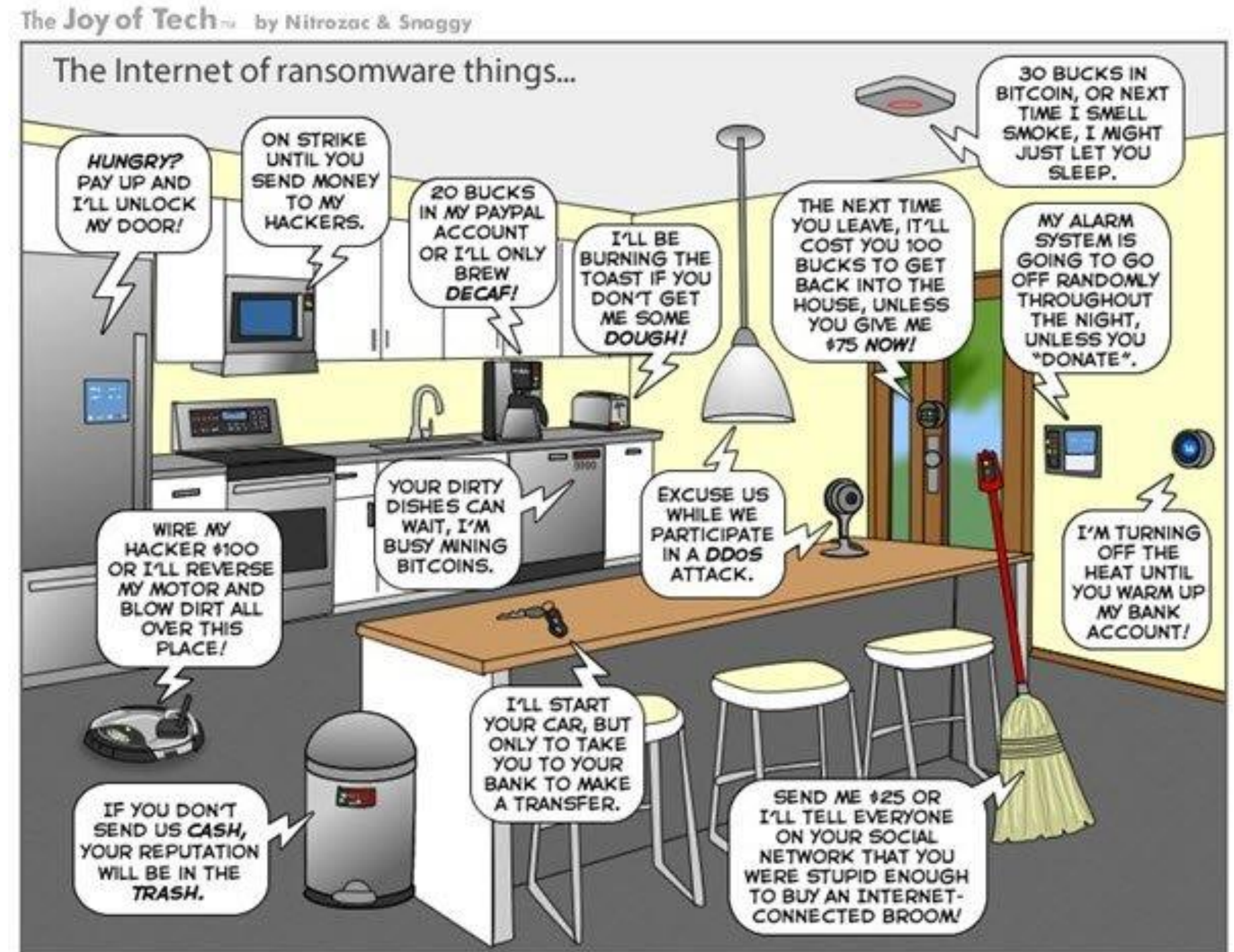
- Complexity
- Interactions
- Weaknesses

of all the layers and components if we want to make this **secure**

IoT Security

The “S” in IoT stands for Security

- Running joke on the Internet
- Sad but true



You can help us keep the comics coming by becoming a patron!
www.patreon.com/joyoftech

joyoftech.com

Current status of IoT Security

- Most IoT devices are **not secure**
 - Quick time to market and low costs, doesn't leave much room for security
- Many marketplaces, platforms and ecosystems are **not secure**
- Security requirements are **not well defined**, known nor met
- Platform providers and end-users cover the **cost of security**
 - Pay for the audits
 - Risk reputation damage
 - Suffer the attacks

Current status of IoT Security

Consumer IoT

- Cheap devices
- Quick time to market
- Disposable devices



Enterprise or Industrial IoT (OT)

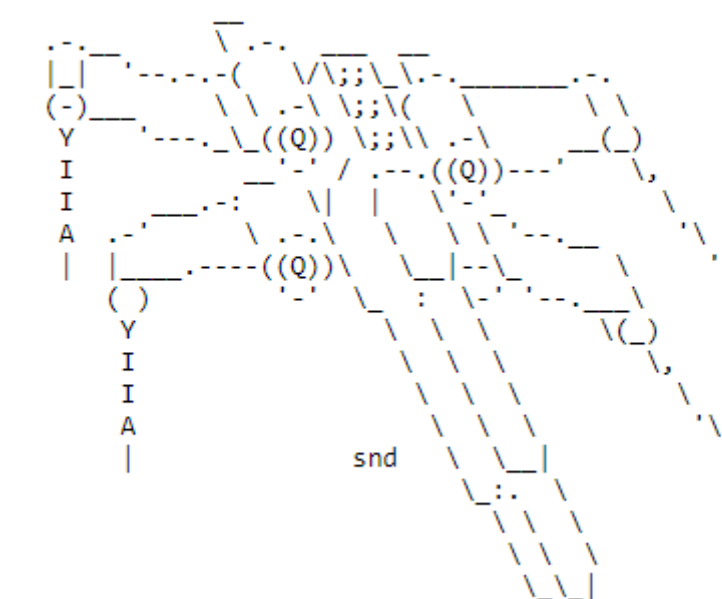
- Proprietary or binary protocols
- ~~Old~~ Operational Technologies (OT)
- “Just transmitting sensor data”
- “We only use network technology X”, so we are secure
- “All our communication take place behind our firewalls”
- In most cases these are false premises

Current status of IoT Security

- We are **recycling bugs and errors** from the past
 - It's like a 90s come back
- As an industry, **we should have learned** from these errors already
 - Using clear-text protocols
 - Correct authentication and trust management
 - Non-mature stacks



Art by Shanaka Dias



Why don't we just make them secure-by-design?

- This is the best time to start approaching security
 - Most effective
 - Reduces costs (up to 20-fold)

Secure an existing product

How to secure an existing product?

- There is no easy way to bolt security onto it
 - It will probably take you a couple of releases to be able to do it
- Since there is no magic bullet, how do we do it?
 - A risk-based approach or a full-blown analysis
 - But will require a bigger effort



Security assessment

- Find the attack surface
- Threat modeling / analysis / risk assessment
 - Need some inspiration? -> OWASP IoT Top 10 can help
- Security Testing / Pentesting
- Fix issues
 - Prioritize? -> Quick wins + biggest impact
 - OWASP IoT Top 10 can help with that
- Do a little dance



<https://giphy.com/gifs/dancing-happy-will-smith-bTzFnjHPuVvva>

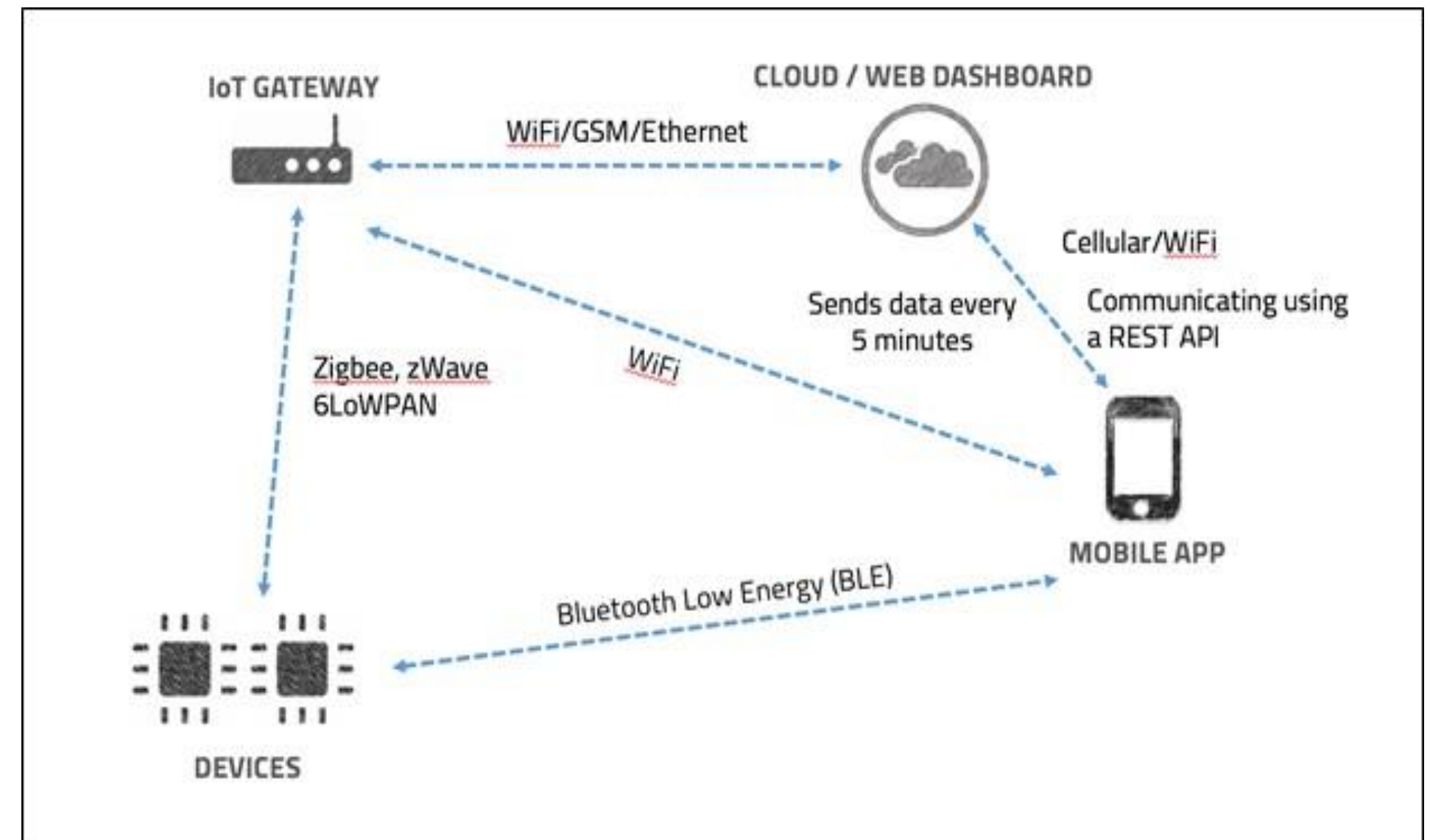
Attack surface for IoT devices

Can be split into 4 categories :

- Device vulnerabilities
- Firmware based vulnerabilities
- Mobile, Web and Infrastructure, and Network security issues
- Radio communication-based vulnerabilities

Threat analysis

- Always look at the big picture
- Create a design diagram
 - Identify all components
 - Identify all interactions between them
- Use your favorite methodology
 - STRIDE
 - VAST
- This can take a while



Fixing issues

- It is hard to define the priorities
- Common approach:
 - Go for the quick wins first
 - Example: Jeep and enabling client isolation
 - Highest impact issues second
 - OWASP IoT Top 10 comes in handy

Short cut: Use the OWASP IoT top 10

- Just test and review the OWASP IoT Top 10

Disclaimer



This does not replace a regular security process

- But it can provide you with many quick wins with a big impact

OWASP IoT Project

OWASP IoT Project

“The OWASP Internet of Things Project is designed to help manufacturers, developers, and consumers better understand the security issues associated with the Internet of Things, and to enable users in any context to make better security decisions when building, deploying, or assessing IoT technologies.”

OWASP IoT Top 10

I1. Weak Guessable, or Hardcoded Passwords

I2. Insecure Network Services

I3. Insecure Ecosystem Interfaces

I4. Lack of Secure Update Mechanism

I5. Use of Insecure or Outdated Components

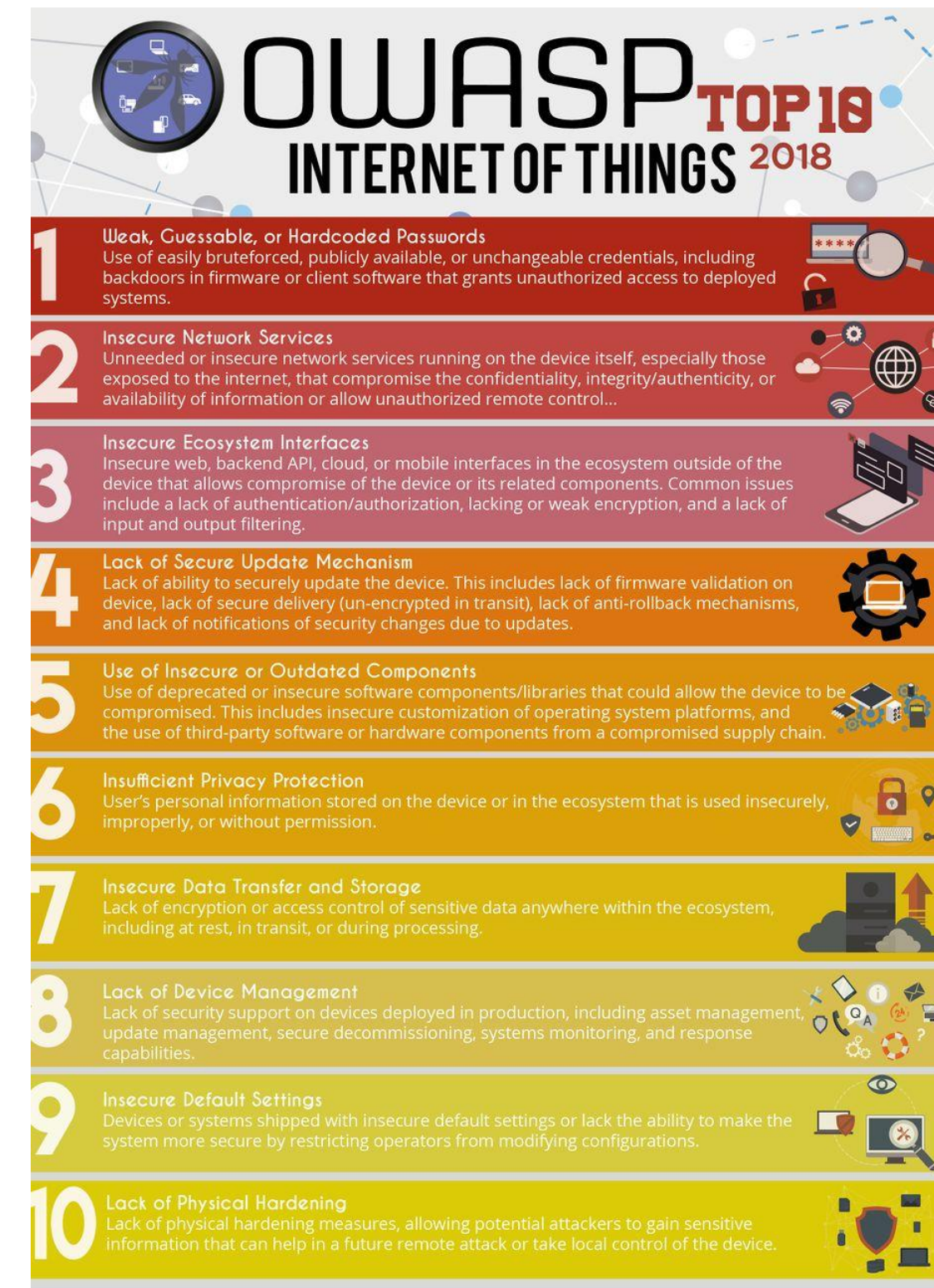
I6. Insufficient Privacy Protection

I7. Insecure Data Transfer and Storage

I8. Lack of Device Management

I9. Insecure Default Settings

I10. Lack of Physical Hardening



Use Cases

Use case: GPS Tracker

Device:

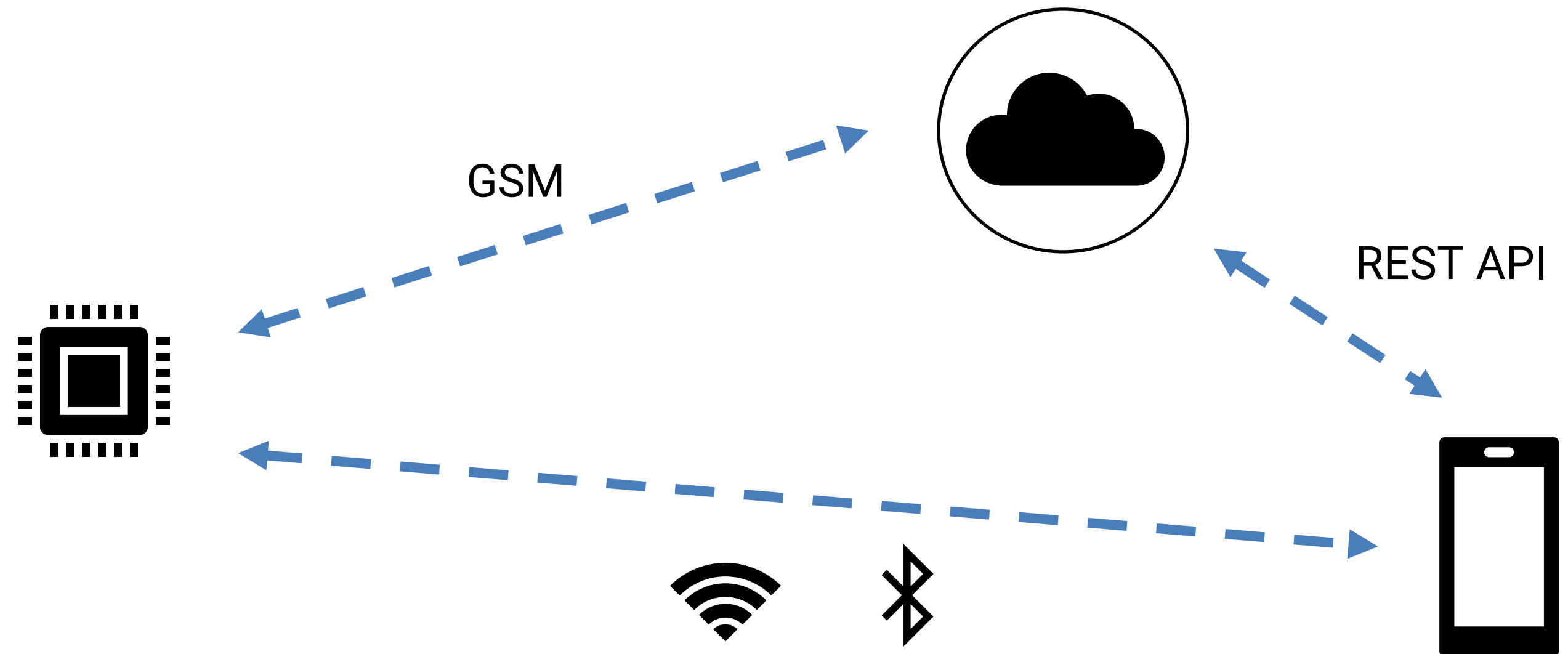
- GPS
- WiFi
- Bluetooth
- GSM (2G / 3G)

Backend:

- REST API
- HTTPS

Mobile App

- HTTPS



Use case: GPS Tracker

Mobile app

- Certificate validation and pinning
 - Dictionary and brute-force attacks
 - Authorization
 - Business and Logic flaws
 - Hardcoded sensitive information
 - Outdated and /or insecure 3rd party libraries and SDKs
-
- Trackers lots of trackers

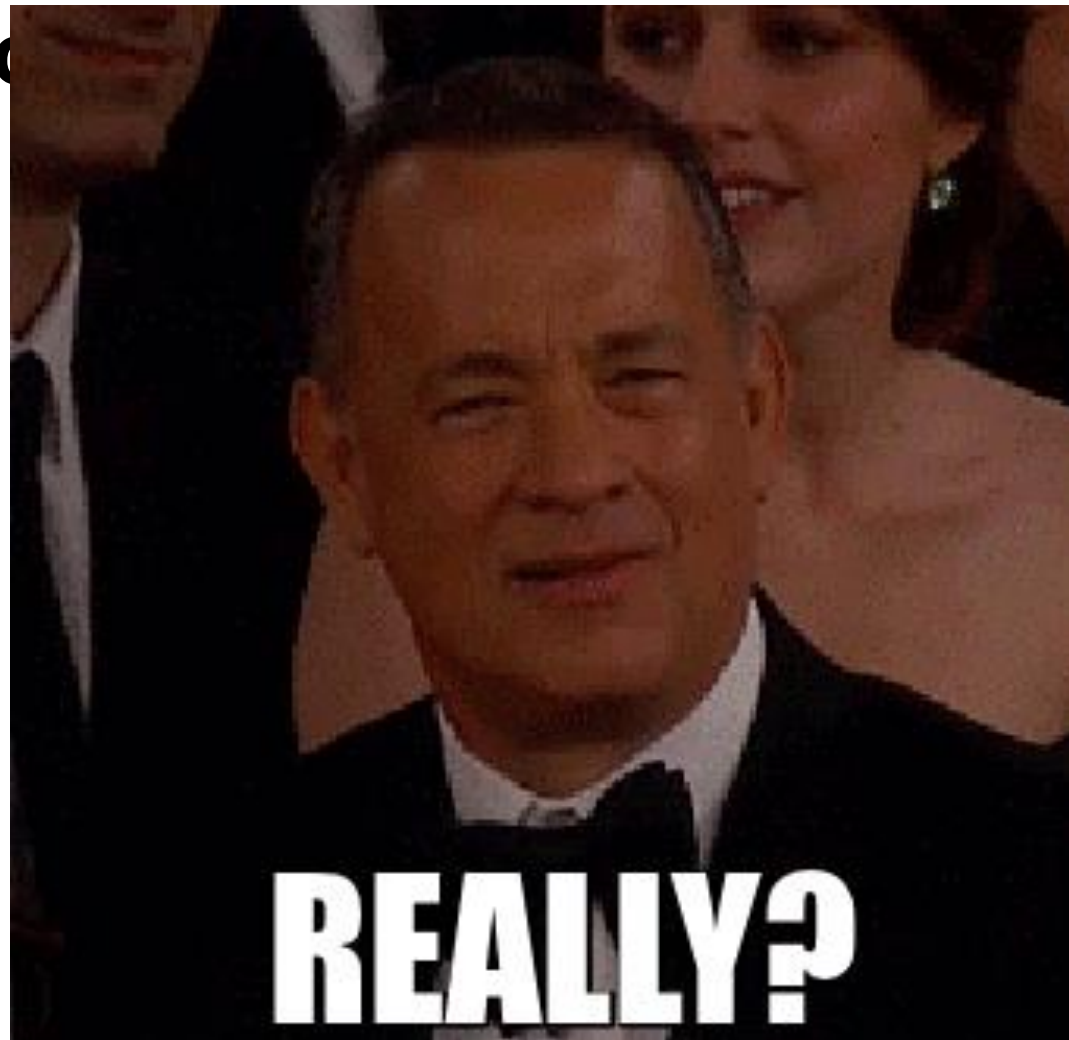
Web portal

- Insecure SSL setup
- Insecure authentication and authorization
- Injection flaws
- Outdated and /or insecure 3rd party libraries and SDKs

Use case: GPS Tracker

Clo

- on
- sitive resources
- data
-
-

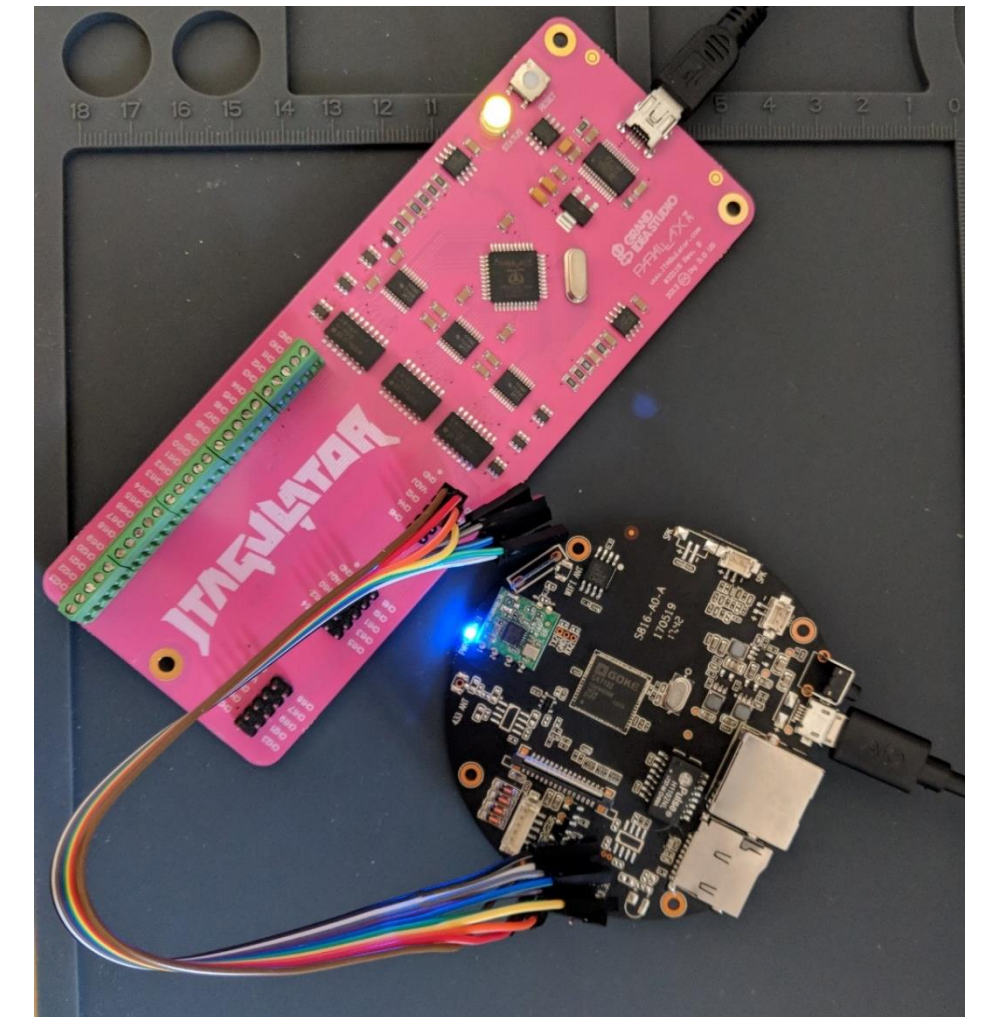
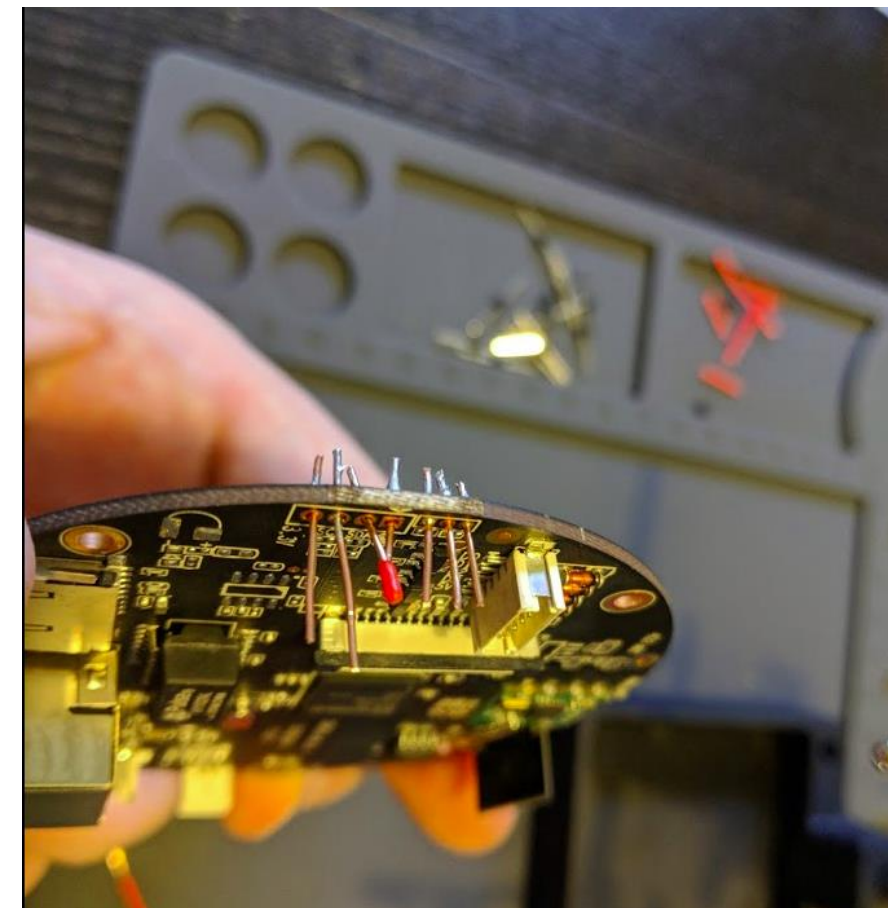


<https://media.giphy.com/media/oOTTyHRHj0HYy/giphy.gif>

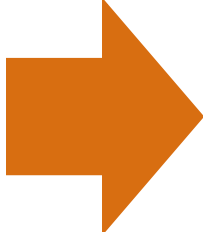
Use case: GPS Tracker

Device + Firmware

- Exposed debugging or serial interfaces
- Ability to dump sensitive information or firmware from flash chips
- Insecure integrity and signature verification
- Insecure OTA update mechanism
- Plain-text traffic



Status: Consumer IoT

- Not secure-by-design (nor implementation)
- Affected by all OWASP Top 10s:
 - Mobile applications
 - Web Applications and APIs
 - IoT
- Biggest issues are in the back-end and APIs (affecting all customers)
- Usually require 4 iterations to pass  Takes 6 to 9 months
 - Is too long for a start-up

Use case: Smart City irrigation system

Devices:

- Sensor / Actor
- Zigbee

Gateway:

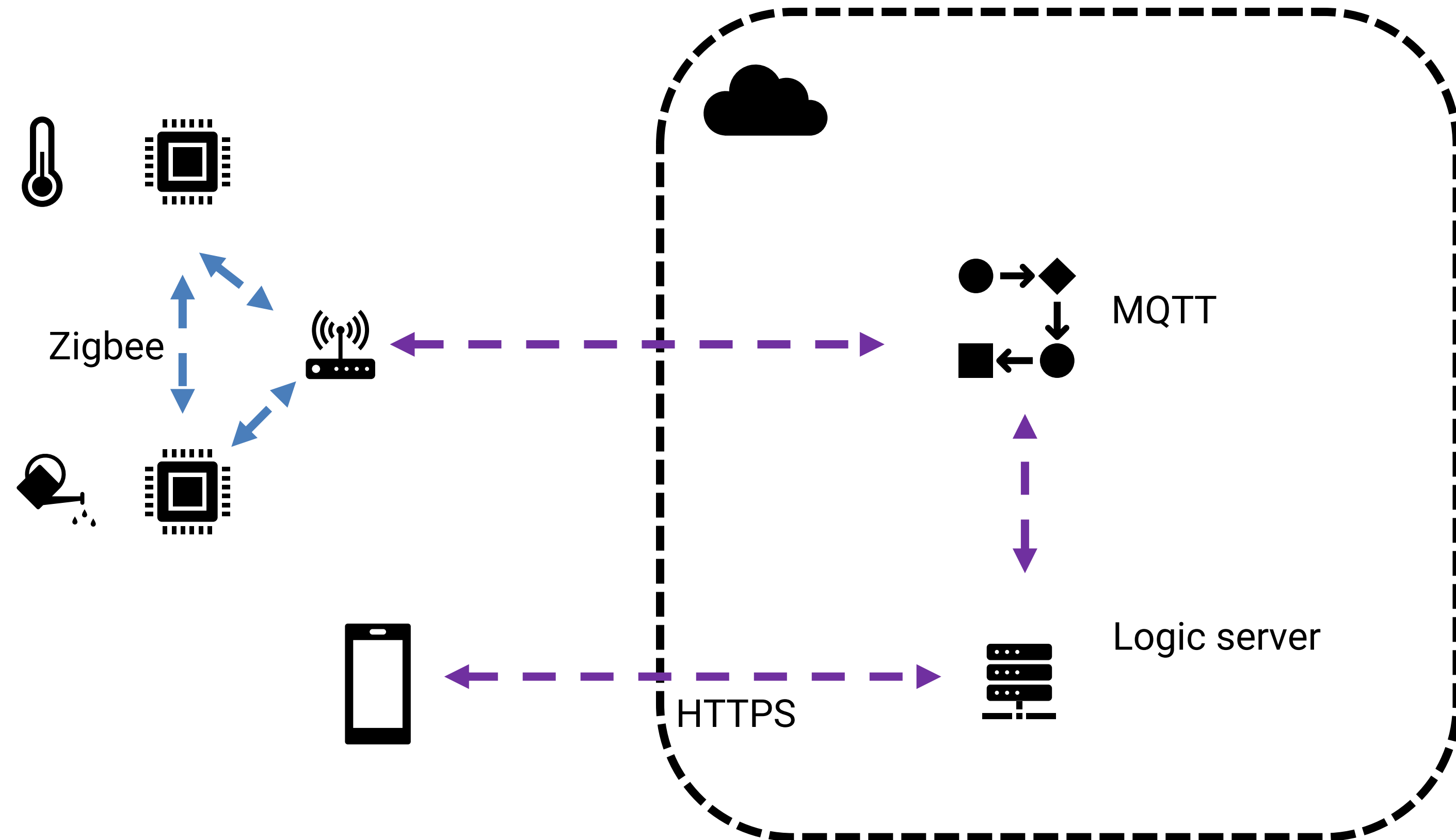
- Zigbee
- Wi-Fi / LAN

Backend:

- MQTT Queue
- REST API
- HTTPS

Mobile App

- HTTPS



Use case: Smart City irrigation system

Wireless

- Did you enable encryption?

MQTT

- Insecure network communication
- Insecure authentication and authorization
- Privilege escalation

Use case: Smart City irrigation system

Device + Firmware

- Insecure removable media
- Exposed debugging or serial interfaces
- Ability to dump sensitive information or firmware from flash chips
- Insecure integrity and signature verification
- Insecure OTA update mechanism
- Plain-text traffic

Cloud Service (API)

- Insecure API communication
- Improper protection of sensitive resources
- Ability to modify sensitive data
- Injection based attacks
- Exposed S3 Buckets

Use case: Smart City irrigation system

Mobile app

- Insecure network communication
- Insecure authentication and authorization
- Business and Logic flaws
- Hardcoded sensitive information
- Outdated and /or insecure 3rd party libraries and SDKs

Web portal

- Insecure network communication
- Insecure authentication and authorization
- Injection flaws
- Outdated and /or insecure 3rd party libraries and SDKs

Status: Smart city

- Large diversity:
 - From Arduino with shields to complex boards with custom firmware
- Main issues
 - Insecure message queues and APIs
 - Lack of authentication and encryption
 - No back-end or API security
 - Wireless: in clear-text or bad encryption. No auth
- Affected by all OWASP Top 10s
- Main justification: “Just transmitting sensor data”



✉ epablo@sevenshift.de

🐦 [@epablosensei](https://twitter.com/epablosensei)

🌐 <https://www.linkedin.com/in/pabloendres>

🌐 <https://pabloendres.com>
<https://sevenshift.de>

Thank you for your time

Feel free to reach out with questions

OWASP IoT Top 10

▪ I1. Weak Guessable, or Hardcoded Passwords

Use of easily brute/forced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed Systems

- Perform brute-force and dictionary attacks
- Extract the firmware
 - Search for strings
 - Decompile if possible, and look for strings or the user auth
 - Google the default passwords for that device



OWASP IoT Top 10

▪ I2. Insecure Network Services

Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control

- Port scans
- Vulnerability scans
- MiTM attacks

OWASP IoT Top 10

▪ **I3. Insecure Ecosystem Interfaces**

Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.

- Normal API / web testing
- MQTT / COAP Testing

OWASP IoT Top 10

▪ **14. Lack of Secure Update Mechanism**

Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.

- Review the upgrade process
- Sniff the traffic
- Reverse engineer

OWASP IoT Top 10

▪ **15. Use of Insecure or Outdated Components**

Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain.

- Retire.js (Burp plugin)
- OWASP Dependency Check
- Black duck / Sonatype / Checkmarx

OWASP IoT Top 10

- **16. Insufficient Privacy Protection**

User's personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.

- Dump EEPROM or other memory chips
- Extract info from mobile apps
- Security testing of mobile apps, APIs and websites

OWASP IoT Top 10

- **17. Insecure Data Transfer and Storage**

Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.

- MiTM
- Follow / Sniff the traffic
- Check certificates
- Review / dump storage

OWASP IoT Top 10

- **18. Lack of Device Management**

Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.

- Ask / typical audit

OWASP IoT Top 10

▪ **19. Insecure Default Settings**

Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

- Test passwords
- Port scans i.e. nmap
- Vulnerability scans

OWASP IoT Top 10

- **I10. Lack of Physical Hardening**

Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

- Take device apart
- Check device for interfaces
- JTAGulator / JTAGenum
- BusPirate