

CT106-3-M-BIA - Building IoT Applications

CT127-3-M-ODL - BIA - Building IoT Applications

## **Topic 8 - IoT Security & Privacy**

# TOPIC LEARNING OUTCOMES

At the end of this topic, you should be able to:

- Articulate the reasons why security is crucial in IoT deployments.
- Comprehend the principles of securing IoT networks.
- Explore the significance of managing updates in IoT security.
- Understand the defence-in-depth strategy for securing IoT applications.
- Apply security principles to real-world IoT scenarios.

# Contents & Structure

- Security by design
- Cloud Processing and Storage
- Securing the network
- Manage secure updates
- Security is important
- Defence-in-Depth Strategy

# Examples of security issues

## Door locks

- cars
- house

## House appliances

- burn
- used as network bots

## Medical devices

- harm people

## Public utilities

- power grid
- water network

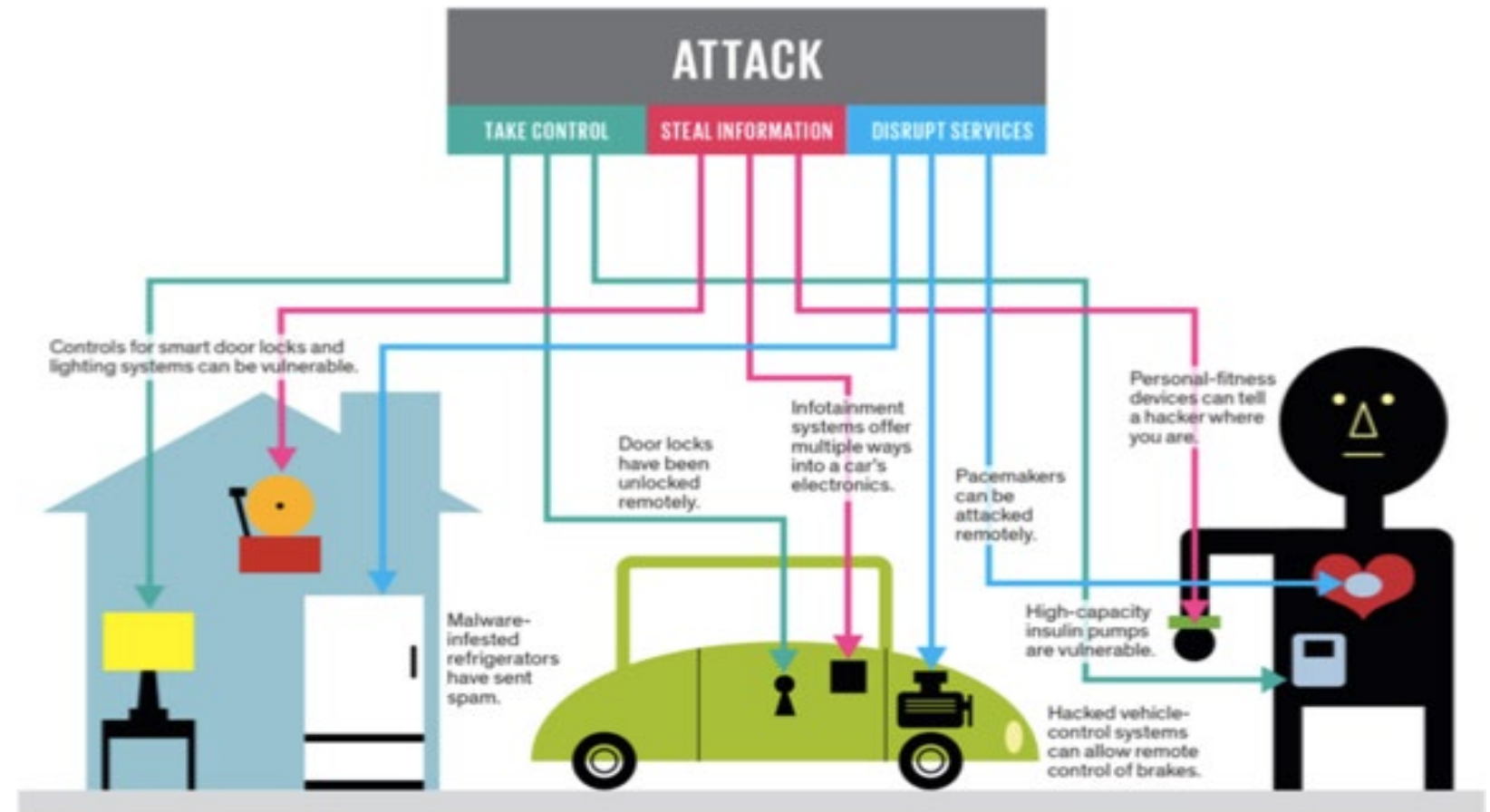


Illustration: J. D. King

<https://www.pubnub.com/blog/2015-05-04-10-challenges-securing-iot-communications-iot-security/>

# Security types

## SECURITY BY DESIGN

Theoretically proven

Usually open solution

Tested and reviewed by a large number of users

Trust the users

## SECURITY BY OBSCURITY

Closed box

No one knows what is inside

If hacked, all the systems fail

How is it updates

Trust the enterprise that designed it

# Securing a device

Local Security

Network Security

Software

Hardware



# Change the default password

Raspberry Pi

- pi/raspberry

BeagleBone

- debian/temppwd

Mirai Net

- Rent devices for DDoS

Distribute devices with a random default password





# Disable unused services

## SSH

- login access

If you don't need it, stop it!

## X Server

- UI, unless you have a display
- default login

Disable administration over the air

## Avahi

- device discovery (multicast)

## SMB (Samba)

- **WannaCry, used SMB 1**



# Avoid self-written protocols

You are the the only one using it

No one tested it

Is it theoretically secure?

Firewalls might stop it

# Use secure protocols

## HTTPS

- Authenticates the server
- Encrypted communication

## MQTT/SSL

- Encrypted MQTT

## XMPP

- Secure messages exchange protocol
- Authenticates servers between each other

## Devices

- Computers
- Microcontrollers



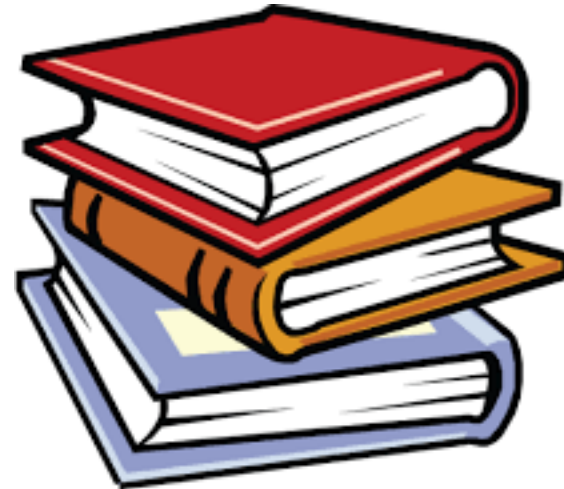
RASPBERRY

# Read before implementation

Read about security issues in the field

Study what experts in the field say

Understand the security problems







# What should you use?

The right hardware for the right job

Use hardware that is able to secure the network

Follow the IoT stack



# Microcontrollers and Computers

- Simple systems Control hardware Low speeds Small memory
  - 2 KB
  - RSA key might be is 2KB
- Run single software
  - RTOS
- Local network only



Full CPUs

High speeds

Large memory

- Is able to use security

Run OS

- Linux OS

Local network and Internet







# Supported Software

---

Raspberry Pi

Arduino YUN

BeagleBone

Arduino TIAN

UDOO

CHIP

Banana Pi



CHIP

*The World's First  
\$9 Computer!*



# Upstream changes

If you change software, push it upstream

For every software update, you have to port your software for it



# Use open libraries

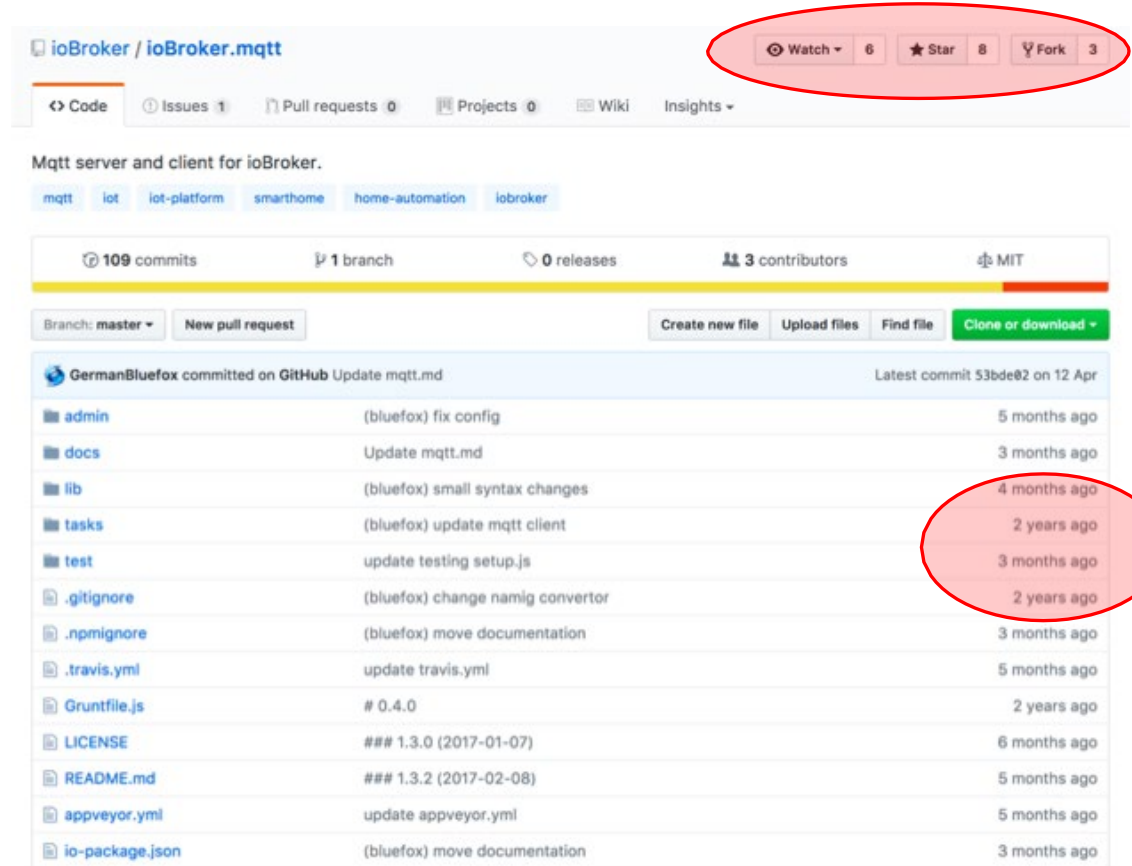
If the protocol is used, someone wrote a library

Use a library that is actively maintained

Follow security updates for the library



# Sure you want to use it?



ioBroker / ioBroker.mqtt

Watch 6 Star 8 Fork 3

Code Issues 1 Pull requests 0 Projects 0 Wiki Insights

Mqtt server and client for ioBroker.

mqtt lot lot-platform smarthome home-automation iobroker

109 commits 1 branch 0 releases 3 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

GermanBluefox committed on GitHub Update mqtt.md Latest commit 53bde02 on 12 Apr

admin	(bluefox) fix config	5 months ago
docs	Update mqtt.md	3 months ago
lib	(bluefox) small syntax changes	4 months ago
tasks	(bluefox) update mqtt client	2 years ago
test	update testing setup.js	3 months ago
.gitignore	(bluefox) change namig convertor	2 years ago
.npmignore	(bluefox) move documentation	3 months ago
.travis.yml	update travis.yml	5 months ago
Gruntfile.js	# 0.4.0	2 years ago
LICENSE	### 1.3.0 (2017-01-07)	6 months ago
README.md	### 1.3.2 (2017-02-08)	5 months ago
appveyor.yml	update appveyor.yml	5 months ago
io-package.json	(bluefox) move documentation	3 months ago

# This is all right to use

mqttjs / MQTT.js

Watch 170 Star 2,357 Fork 457

Code Issues 40 Pull requests 7 Projects 0 Wiki Insights

The MQTT client for Node.js and the browser

mqtt javascript nodejs-library mqtt-broker

1,057 commits 21 branches 104 releases 97 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

mcollina Bumped v2.9.1. Latest commit #b6f6c4 23 hours ago

benchmarks	Standardized	10 months ago
bin	remove 'integer' argument to minimist	8 months ago
doc	add --multiline to pub.js for message streaming via stdin	a year ago
examples	fix retain flag in publish, was retained typo	3 months ago
lib	use xtend instead of extend	a day ago
test	the default value must be set for an empty options parameter	2 days ago
types	fix type of some fields in ISecureClientOptions	2 months ago
.editorconfig	Various tweaks:	4 months ago
.eslintrc	removed unnecessary comments and corrected styling for new eslint	2 years ago
.gitignore	Fix travis & Add coverage.io	5 months ago
.jscsrc	Merge branch 'dev-jshint' of https://github.com/itavy/MQTT.js into it...	2 years ago
.jshintrc	added rules for jshint and jscs	2 years ago
.travis.yml	Added node 8 to .travis.yml	a month ago

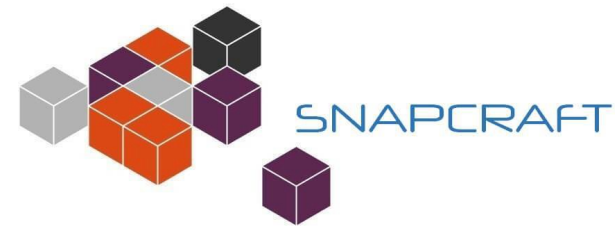
# How do you update the device?

Your software will have update

- features
- Security

OS

- dual partition



Applications

- snap
- Google Store (Android Things)





# Trusted software

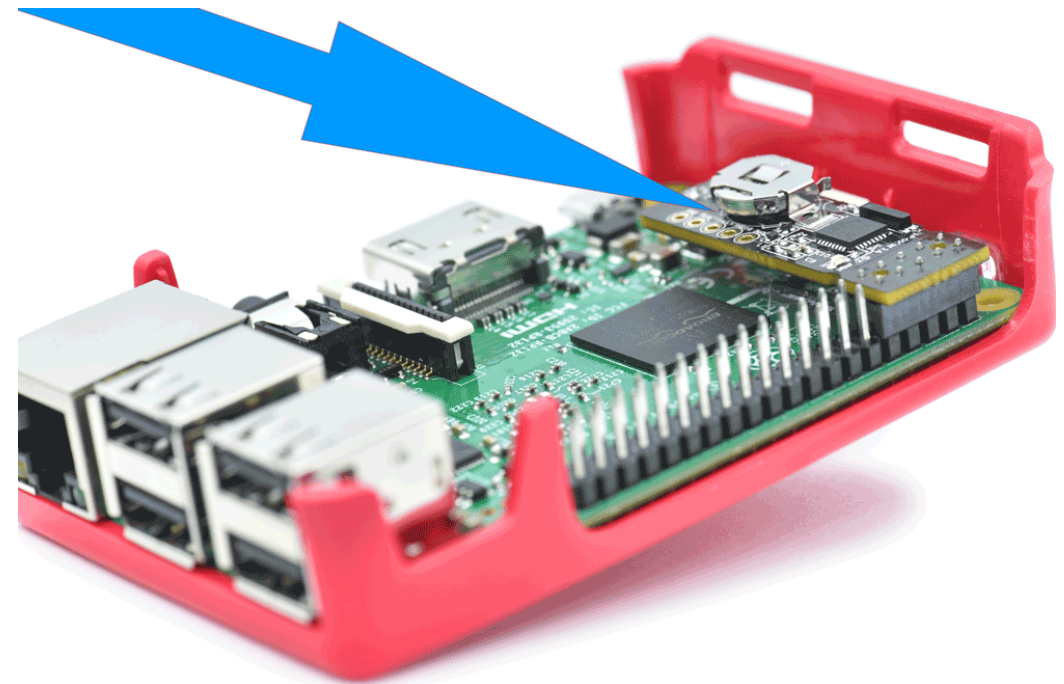
Digitally sign the software

Secure boot

- hardware support here
- additional hardware

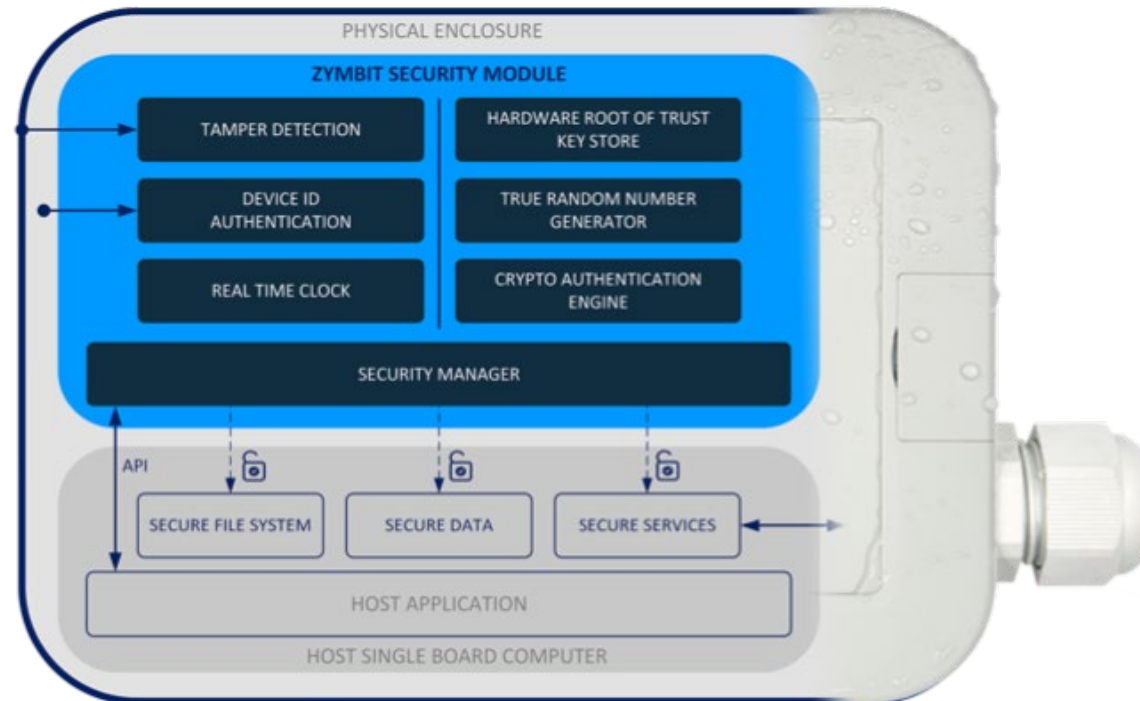
Secure software

- digitally signed
- Verified before install





# Trust store



# Defence-in-depth

Defence-in-depth involves implementing security measures at various IoT levels, including the device, network, and application layers. It involves implementing multiple layers of security measures to protect IoT devices and networks that involve:

- **Device-level security:** This layer focuses on securing individual IoT devices. It includes measures such as strong authentication mechanisms, secure boot, tamper-proofing, and device-level encryption. By implementing these measures, the device itself is protected from unauthorized access and tampering.
- **Network-level security:** This layer focuses on securing the communication between IoT devices and the network infrastructure. It involves implementing secure communication protocols, such as Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS), to encrypt data transmission. Additionally, network-level security measures include network segmentation, firewalls, intrusion detection and prevention systems, and virtual private networks (VPNs) to protect against unauthorized access and network-based attacks.

# Defence-in-depth

- **Application-level security:** This layer focuses on securing the applications and services running on IoT devices. It involves implementing secure coding practices, input validation, and access controls to prevent common vulnerabilities, such as buffer overflows and injection attacks. Application-level security measures also include regular patching and updates to address known vulnerabilities.
- **Data security and privacy:** Defense-in-depth also emphasizes protecting the sensitive data collected and transmitted by IoT devices. This involves implementing encryption techniques, access controls, and data anonymization methods to ensure data confidentiality and integrity. Additionally, compliance with privacy regulations and secure data storage practices are essential.

By implementing multiple layers of security measures, defence-in-depth ensures that even if one layer is breached, there are additional layers in place to mitigate the risks and protect the IoT ecosystem. It creates a more robust and resilient security posture by reducing the likelihood of successful attacks and minimizing the potential impact of a security breach.

# Review Questions

- What are the challenges and vulnerabilities associated with securing IoT devices? Explain how these challenges differ from traditional computing systems.
- Discuss the concept of defence-in-depth in the context of IoT security. How does it involve implementing multiple layers of security measures to protect IoT devices and networks?
- Discuss the key aspects of network security, such as encryption, authentication, and access control, in the context of IoT.
- Explain the importance of secure communication protocols in IoT. Discuss the role of protocols like Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) in ensuring secure data transmission.

# What To Expect Next Week

In Class

Preparation for Class