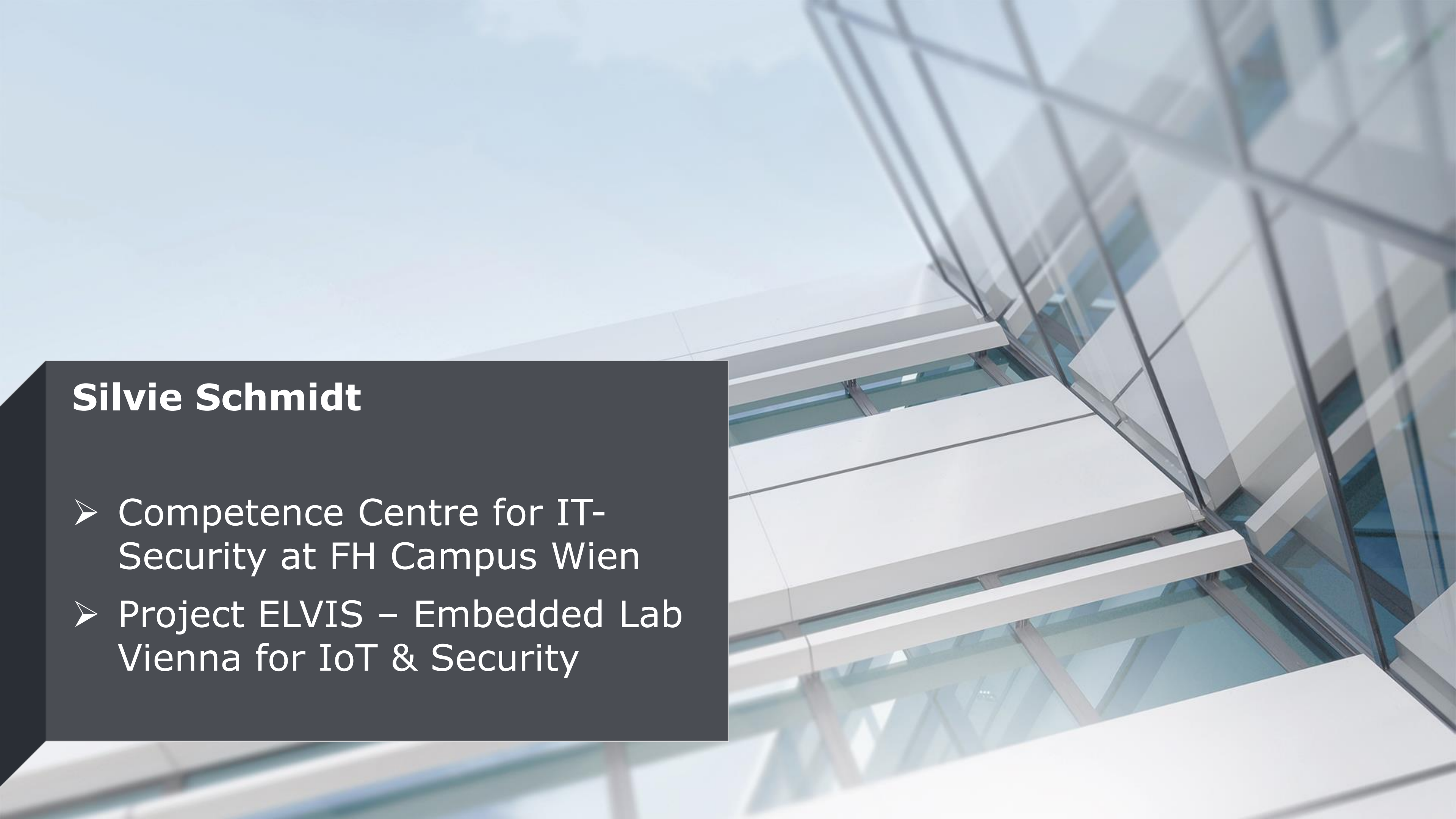
FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

DIE HOCHSCHULE,
DIE MEHR KANN.

Secure Firmware Updates in the IoT

COMPETENCE CENTRE FOR IT-SECURITY, MASTER STUDIES IT-SECURITY



Silvie Schmidt

- ➢ Competence Centre for IT-Security at FH Campus Wien
- ➢ Project ELVIS – Embedded Lab Vienna for IoT & Security

Agenda

- Requirements, Threats
- Common Strategies
- Recent Projects

- Live Demo – Riot-OS SUIT Example

- Please check the last two slides for sources used in this presentation (figures, etc.)

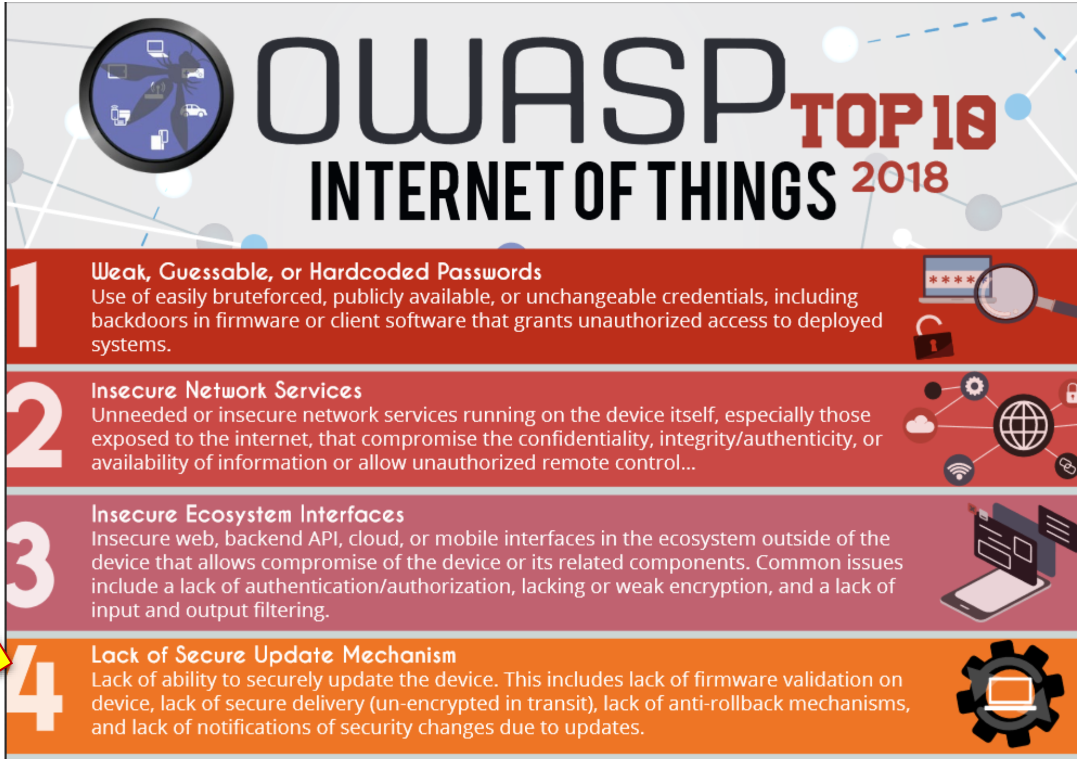
The Firmware Update Process ….

> …is crucial in the Internet of Things

> …and one of the most critical processes

[p.40(11)]

p.40(10)



OWASP TOP 10
INTERNET OF THINGS 2018

1 **Weak, Guessable, or Hardcoded Passwords**
Use of easily bruteforced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.

2 **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…

3 **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.

4 **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.

Definitions

- Constrained devices: no common OS, embedded OS, e.g. Contiki, RIOT-OS,…

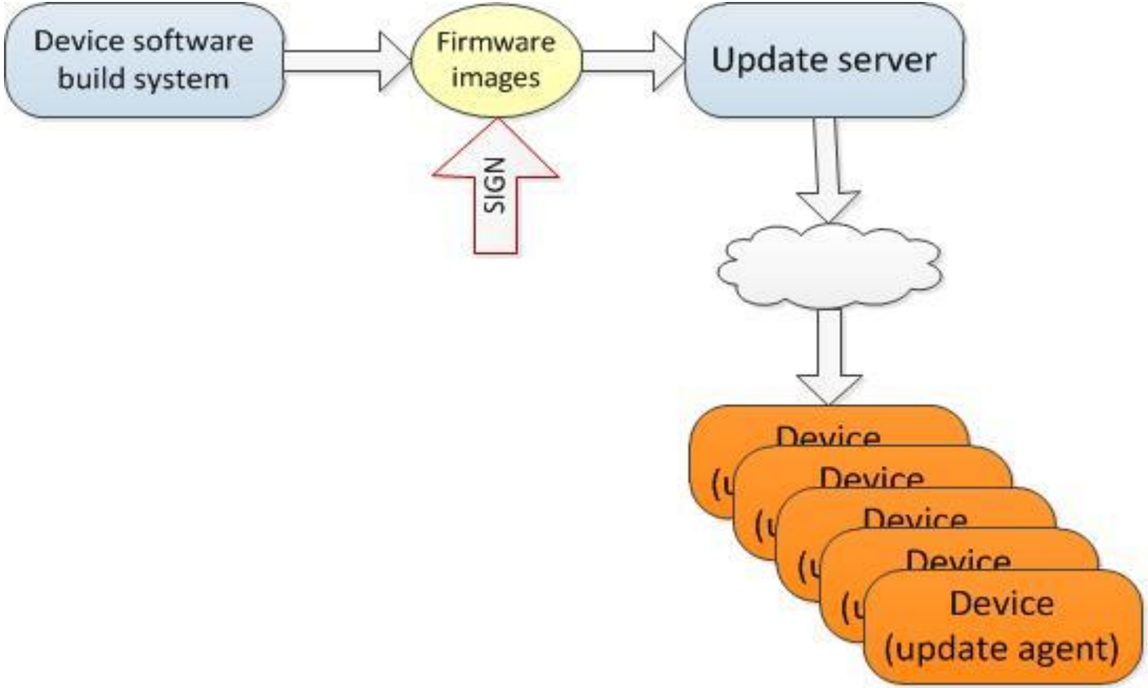
- Firmware:
  - IEEE: combination of HW & SW
  - Often: either exclusively HW or SW
  - In this talk: application that runs on the device (SW)

- FOTA: Firmware update over the air

Why is Firmware Updated?

- Bug fixes
- New features
- Security patches
- ….

FOTA Components



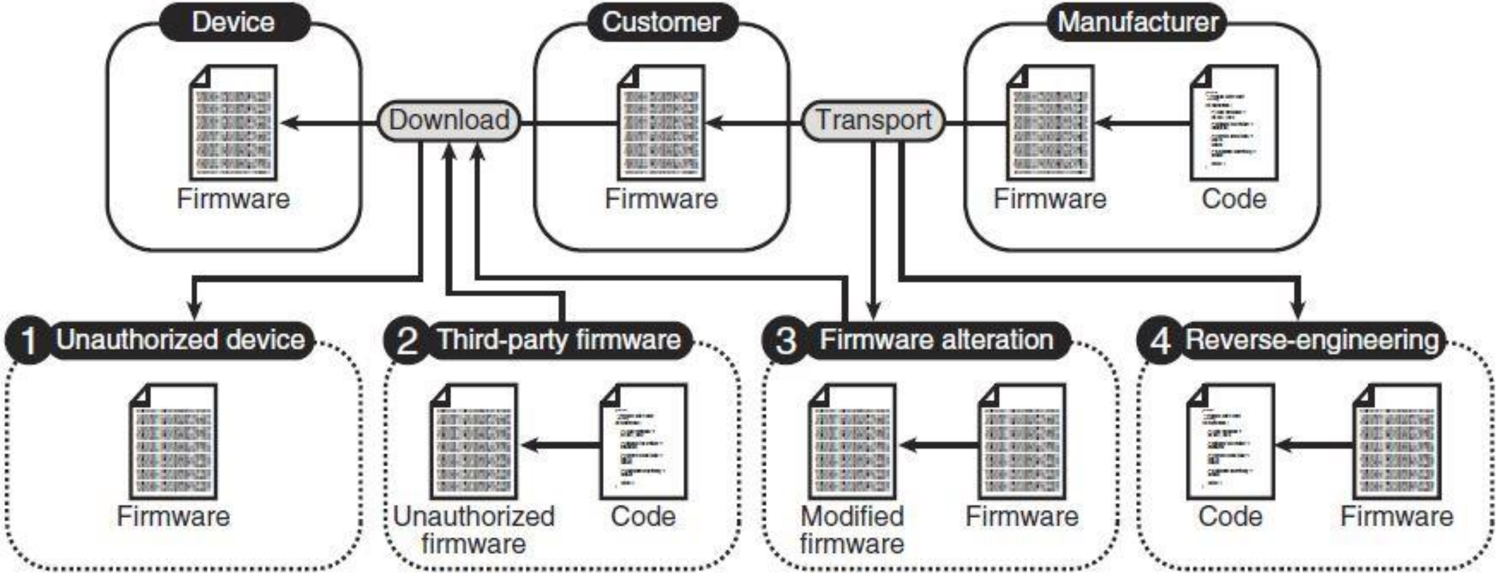
Based on [p.39(4)]

Threats

- ***What can go wrong?***
  - Wrong firmware
  - Bad firmware
  - Power failure
  - Transmission errors
  - Not working firmware
  - And many more….

Threats

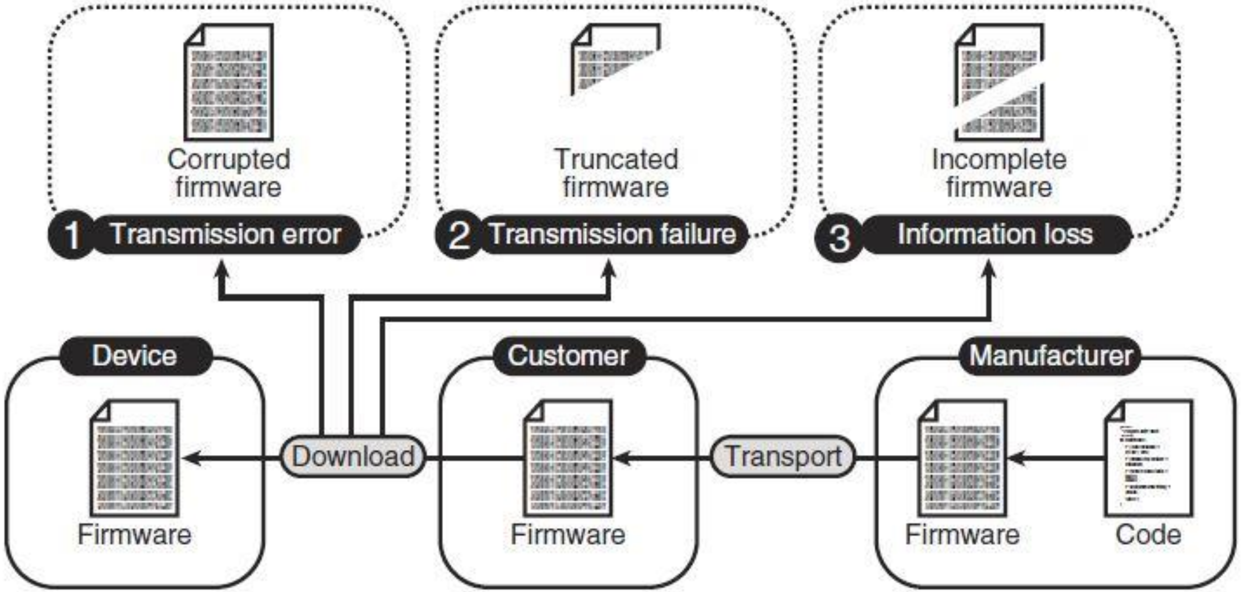
> *Update Process Security Issues*



[p.39(3)]

Threats

> *Update Process Safety Issues*



[p.39(3)]

Requirements

- ***Main Requirements for a Secure FW Update***
  - Security
    - Prevent hijacking
  - Robust
    - Update may not cause a broken device
  - Atomic
    - All or nothing
  - Fail-safe
    - Roll-back mode

Firmware Integrity

- ***Most used security feature***
  - Often the only implemented security feature
    - Each additional security feature decreases performance by any means
  - Integrity techniques solve many security issues:
    - Recognition of tampered, wrong, and incomplete images
    - Transmission errors (both, (un)intentionally)
    - Recognition of information loss
  - BUT not everything is solved

Security Requirements

- ***Considerations***
  - Device
  - Scope of application
  - Performance
  - Energy
  - …

Security Requirements

- ***Example***
  - Authentication
  - Version control
  - Code integrity
  - Complete & error-free transmission
  - Operability check
  - Reduced user interaction

Besides Security

- ***Considerations***
  - Update process initiated by the server or by the client?
  - Necessary frequency of the firmware updates
  - Does each device receive the same update image?
  - Do all devices need an update?
  - ….

Security

- ***Conclusion – for now***
  - In general, stronger security results in weaker performance!
  - Basis for trade-off: application scenario

Firmware Update Strategies

- In general, a FOTA in the Internet-of-Things (IoT) is done by replacing the full firmware at once (for simplicity reasons).

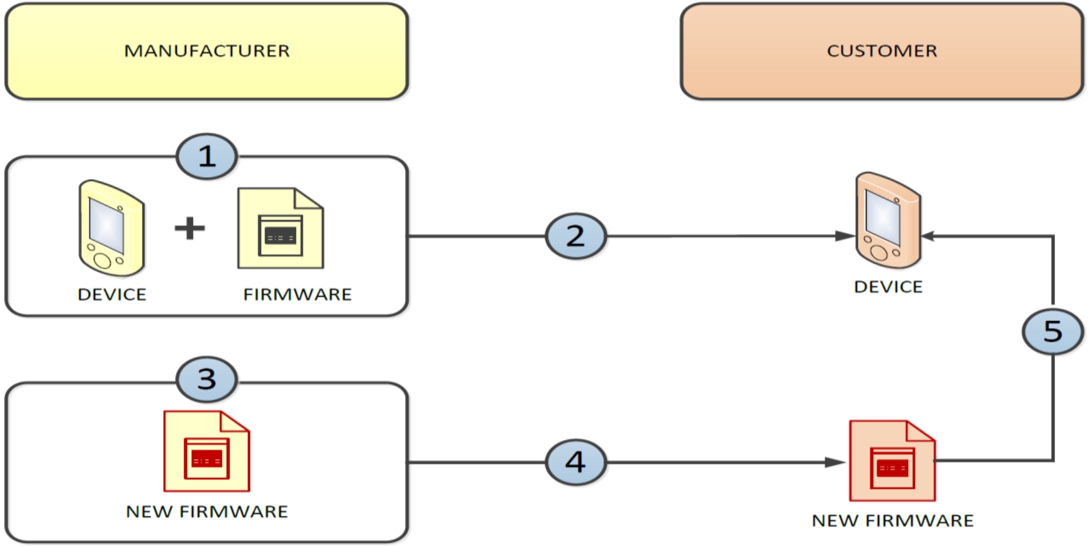
- Nevertheless, there are more options, i.e. strategies.

Firmware Update Strategies

- ***Steps of a Firmware Update Process (example)***
    - Initialization via client or server
    - Transmission of the new firmware image
    - Validation of the update image‘s integrity
    - Decryption of the update image
    - Operational tests
    - …

Firmware Update Strategies

> *Infield Updates*



Based on[p.39(3)]

Firmware Update Strategies

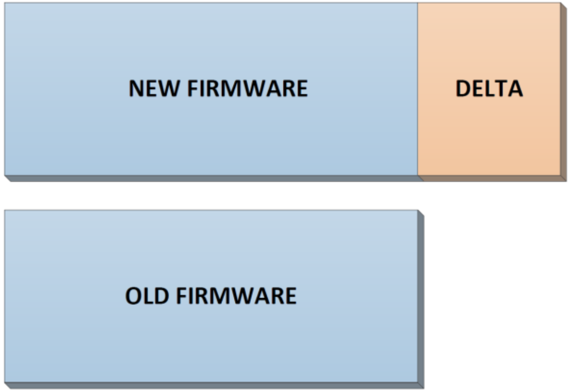
- ***Infield Updates***
  - Manufacturer designs device & firmware
  - Devices with firmware sold
  - New version of firmware developed
  - Distribution to customers
  - Customers patch devices

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Firmware Update Strategies

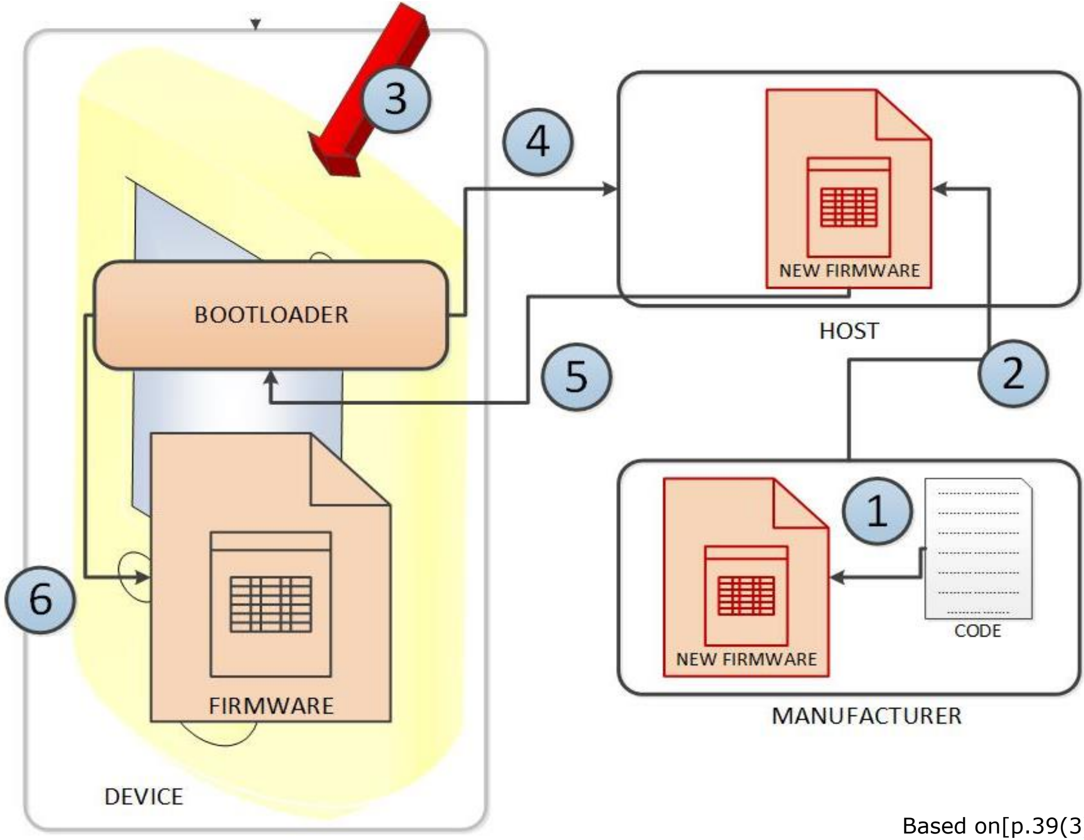
> *Incremental FW Updates*

- > Focus on decreasing transmitted data
- > Code delta is updated (e.g. libraries)



FWU Strategies

> ***Bootloader-Based FWU***



Based on[p.39(3)]

Firmware Update Strategies

- ***Bootloader-Based FWU***
  - After distribution to users boot condition is triggered
  - FWU transmission
  - Old FW replaced by new one
  - New FW started

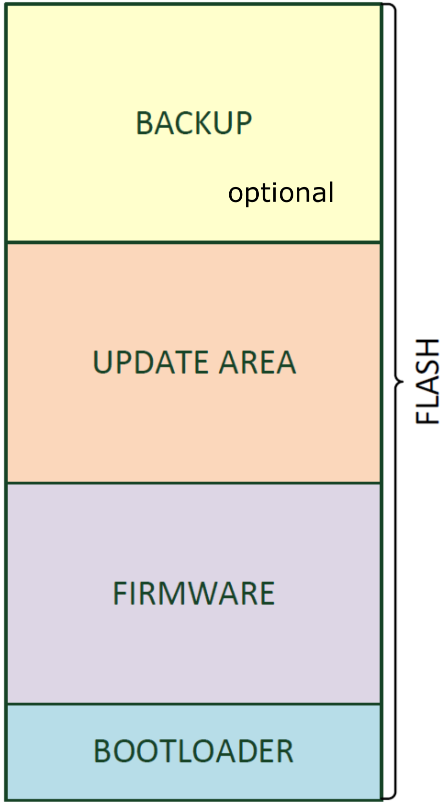
Firmware Update Strategies

- ***Bootloader-Based FWU cont‘d***
  - Trigger conditions:
    - Hardware, e.g. reset button
    - Software, e.g. no valid application
  - On system start the bootloader checks the predefined conditions

Firmware Update Strategies

> ***Memory Partitioning***

- > Solves all safety issues
- > Needs extra memory
- > Always a working firmware available



Conclusion for FWU Strategies

- Secure FW updates in the IoT are not trivial
  - The software on the devices needs to be prepared to support a FW update mechanism
    - E.g. a bootloader which determines which firmware to launch
    - Furthermore, the bootloader executes cryptographical operations like signature verification, decryption, etc.
    - Lastly, the bootloader may also do operational checks for the new firmware
    - Memory layout has to be considered (various slots, e.g. bootloader, application, update area)

IoT Device Management

- ***Open Source Standards for Remote IoT Device Mgmt***
  - LWM2M: OMA, may be secured with DTLS [p.40(4)]
  - CoMI: IETF, CoAP Management Interface [p.40(5)]
  - OCF: Open Connectivity Foundation (CoAP, TLS/DTLS) [p.40(6)]
  - TR69 protocol: broadband forum, most used IoT management protocol [p.40(7)]

Firmware Update Frameworks

> SUIT – IETF working group for SW updates in the IoT (successor of FOSE)[p.40(1)]
> Uptane, TUF – FWU for connected cars[p.39(11), p.39(7)]
> MCUboot – FOTA for ESP8266 uCs [p.39(6)]
> ReLog, Mate – using miniature VMs[p.39(8), p.39(9)]
> CHAINIAC – blockchain-based [p.40(2)]
> SWUpdate – mainly considered as a framework [p.40(3)]
>

Firmware Update Frameworks

> ***SUIT*** – SW Updates in the IoT

> > IETF working group
> > Simple back-end architecture
> > Authentication & integrity protection
> > Encryption of FW image
> > Secure, even when updates are stored on untrusted repositories

Firmware Update Frameworks

> ***SUIT*** – SW Updates in the IoT

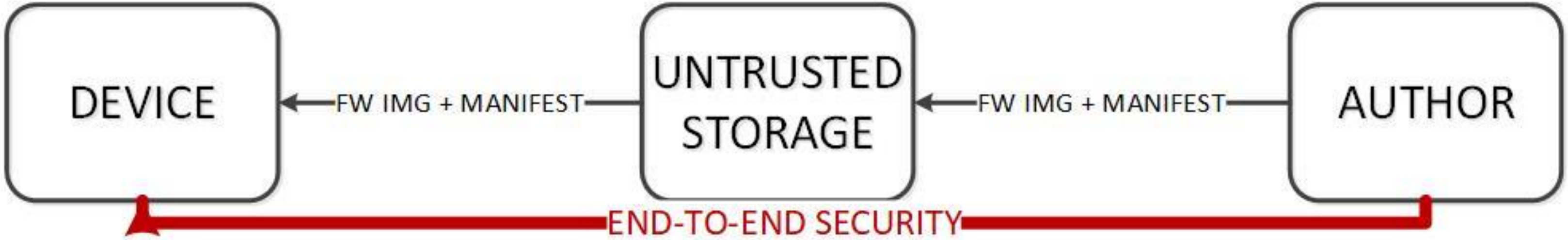
- A manifest standardizes a format for describing FW updates
  - Provides information about the FW required to update device
  - A security wrapper to protect the meta-data end-to-end
  - May provide Uptane-compliant meta-data
- CBOR, COSE
- A firmware update architecture for IoT devices.

Firmware Update Frameworks

- ***SUIT*** – Requirements
  - Agnostic to how firmware images are distributed
  - Friendly to broadcast delivery
  - Use state-of-the-art security mechanisms
  - Rollback attacks must be prevented
  - High reliability
  - Operate with a small bootloader
  - Small Parsers
  - Minimal impact on existing firmware formats
  - Robust permissions
  - Diverse modes of operation
  - Suitability to software and personalization data

Firmware Update Frameworks

- ***SUIT*** – SW Updates in the IoT
  - State-of-the-art security mechanisms
    - End-to-end security between author and device



Based on [p.40(1)]

Firmware Update Frameworks

- ***SUIT*** – SW Updates in the IoT
  - State-of-the-art security mechanisms
    - Mandatory-to-implement set of algorithms with at least keylengths of
      - 112-bit for symmetric cryptography
      - 233-bit for ECC cryptography
      - 2048-bit for RSA

Firmware Update Frameworks

- ***SUIT*** – Manifest contains
  - Information about the device(s) the firmware image is intended to be applied to
  - Information about when the firmware update has to be applied
  - Information about when the manifest was created
  - Dependencies on other manifests
  - Pointers to the firmware image and information about the format
  - Information about where to store the firmware image
  - Cryptographic information such as digital signatures or message authentication codes (MACs)

Firmware Update Frameworks

- ***SUIT*** – SW Updates in the IoT
  - Let‘s take a look at an example: SUIT update with RIOT-OS – the friendly OS for the IoT
  - https://github.com/RIOT-OS/RIOT/tree/master/examples/suit_update

Sources

(1) K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig and E. Baccelli, "Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check," in IEEE Access, vol. 7, pp. 71907-71920, 2019.

(2) K. Doddapaneni, R. Lakkundi, S. Rao, S. G. Kulkarni and B. Bhat, "Secure FoTA Object for IoT," 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), Singapore, 2017, pp. 154-159.

(3) Atmel Application Note AT02333: http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42141-SAM-AT02333-Safe-and-Secure-Bootloader-Implementation-for-SAM3-4_Application-Note.pdf

(4) Chris Simmonds, OpenIoT Summit 2016: https://elinux.org/images/f/f5/Embedded_Systems_Software_Update_for_IoT.pdf

(5) E. Ronen, A. Shamir, A. Weingarten and C. O'Flynn, "IoT Goes Nuclear: Creating a ZigBee Chain Reaction," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2017, pp. 195-212.

(6) MCUboot Project, https://mcuboot.com/

(7) TUF – The Update Framework, https://theupdateframework.io/

(8) Zhu, Xiaorui & Tao, Xianping & Gu, Tao & Lu, Jian. (2016). ReLog: A systematic approach for supporting efficient reprogramming in wireless sensor networks. Journal of Parallel and Distributed Computing. 102. 10.1016/j.jpdc.2016.12.010.

(9) Levis, Philip & Culler, David. (2002). Mate: A Tiny Virtual Machine for Sensor Networks. ACM SIGARCH Computer Architecture News. 30. 10.1145/605397.605407.

(10) Kuppusamy, Trishank & DeLong, Lois & Cappos, Justin. (2018). Uptane: Security and Customizability of Software Updates for Vehicles. IEEE Vehicular Technology Magazine. PP. 1-1. 10.1109/MVT.2017.2778751.

(11) Uptane Project, https://uptane.github.io/

(12) Uptane Design, https://uptane.github.io/design.html

Sources

(1) IETF-SUIT, https://tools.ietf.org/html/draft-ietf-suit-architecture-08

(2) Nikitin, Kirill & Kokoris-Kogias, Eleftherios & Jovanovic, Philipp & Gasser, Linus & Gailly, Nicolas & Khoffi, Ismail & Cappos, Justin & Ford, Bryan. (2018). CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds.

(3) SWUpdate Project, https://sbabic.github.io/swupdate/swupdate.html

(4) LWM2M – Lightweigth M2M, https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/

(5) CoMI – CoAP Management Interface, https://tools.ietf.org/html/draft-ietf-core-comi-04

(6) OCF – Open Connectivity Foundation, https://openconnectivity.org/

(7) TR69 Protocol, https://www.broadband-forum.org/download/TR-069_Amendment-2.pdf

(8) RIOT-OS, https://www.riot-os.org/

(9) https://www.owasp.org/images/1/1c/OWASP-IoT-Top-10-2018-final.pdf

(10)Missbach, N., Secure Firmware Updates for the Internet of ThingsThe IoT, Over-The-Air Updates and possible Solutions, http://pub.fh-campuswien.ac.at/obvfcwhsacc/content/titleinfo/3431921

(11)http://clipart-library.com/clipart/