YAGOBA
[w] www.yagoba.com, [m] info@yagoba.com
Beethovenstraße 20, 8010 Graz, Austria

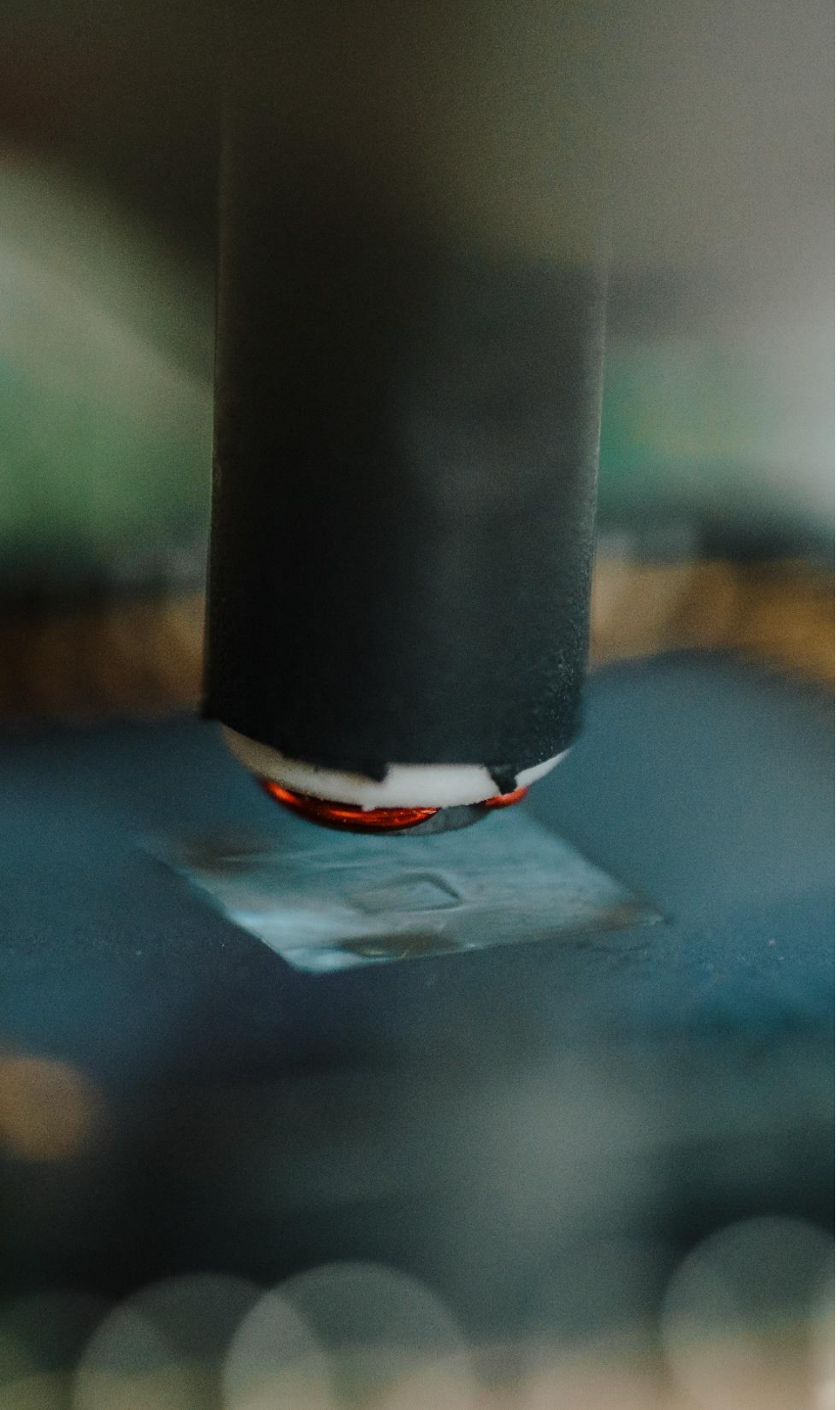
Secure Product Lifecycle

Security from Release to Decomssioning

Christoph Herbst

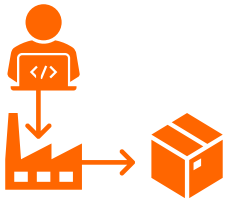
WHEN YOU NEED TO BE SURE

SGS



Guidance and Release

AGENDA



- Release Process
  - Configuration Management
  - Production Security
  - Delivery Security

- Security Guidance
  - Preparational Security Guidance
  - Operational Security Guidance

SECURITY GUIDANCE AND RELEASE MOTIVATION

Secure Release Process

A global cyber-attack that affected companies around the world may have started via corrupted updates on a piece of accountancy software.

Fingers are increasingly pointing to a piece of Ukrainian tax-filing software, MEDoc, as the source of the infection, although the company denies it.

Malware generally infiltrates networks via email attachments that users click on in error.

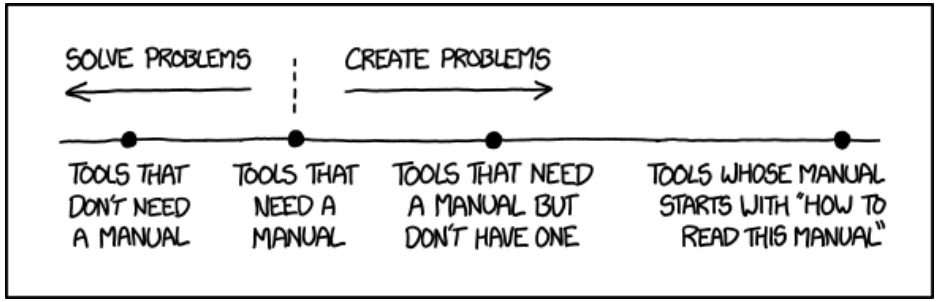
Microsoft described the method as "a recent dangerous trend".

The cyber-attack has caused disruption around the world and infected companies in 64 countries, including banks in Ukraine, Russian oil giant Rosneft, British advertising company WPP and US law firm DLA Piper.

Source: BBC News, https://www.bbc.com/news/technology-40428967

Source: The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies - Bloomberg

Security Guidance



Source: https://xkcd.com/1343/

SECURITY GUIDANCE AND RELEASE MOTIVATION

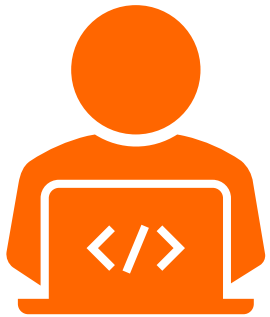
Product Attributes to consider

- Integrity is always an issue
  - attackers who want to weaken or bypass the security
  - missing or insufficient change management
- Confidentiality can be an issue, e.g.
  - Certificates, passwords and similar
  - Known (unpublished) security vulnerabilities
  - Protection of Intellectual Property (IP)
- Authenticity
  - Is the product in hand the one it claims to be?
- Correct Usage
  - Is the product set-up and used in the intended way?

SECURITY GUIDANCE AND RELEASE MOTIVATION

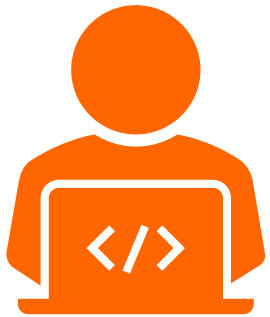
- Development
  - Protect Integrity and Confidentiality of sources and development material
- Production
  - Protect Integrity and Confidentiality of sources and product (parts) during
    - transport between sites and
    - production process itself
- Delivery
  - Protect Integrity of product during storage and delivery
- Security Guidance
  - Provide information to securely set-up product
  - Provide information for secure operation of product

RELEASE PROCESS
CONFIGURATION MANAGEMENT



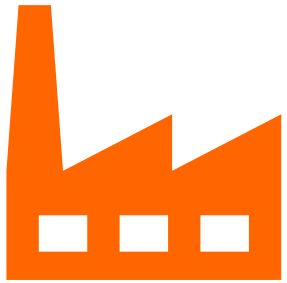
- Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.
- Configuration items can include
  - Source code
  - Design data
  - Documentation (incl. User Guidance)
  - Development Tools
  - Threat Modelling, Risk assessments, etc
- Configuration Management is required by many standards, e.g.
  - ISO 10007:2003 Quality management systems
  - IEEE 829 Standard for Software Test Documentation
  - ITIL Service Asset and Configuration Management
  - ISO 15408 Common Criteria
  - ISO/SAE 21434 Road vehicles – Cybersecurity engineering

RELEASE PROCESS
CONFIGURATION MANAGEMENT



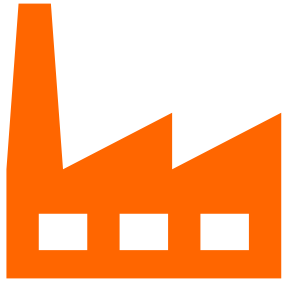
- During development, production and up to release Configuration Management shall
  - ensure that the product is correct and complete before it is sent to the consumer
  - ensure that no configuration items are missed during development, approvals, testing, evaluation etc.
  - prevent unauthorised modification, addition, or deletion of configuration items
- Configuration management (CM) is one means for increasing assurance of correct implementation of the (security) features
  - requires discipline and control in the processes of refinement and modification
  - provides tracking of any changes and ensures that all changes are authorized
  - should ensure the integrity of the product from early design stages through all subsequent maintenance efforts
- Efficiency can be further increased by introducing automated CM tools
  - automated systems are less susceptible to human error or negligence.

RELEASE PROCESS PRODUCTION SECURITY



- Integrity and Confidentiality need to be maintained
  - when configuration items are shipped between different sites (development, production, storage, distribution, etc.)
  - During the production process, especially for HW
  - Can be supported by cross site configuration management systems
- Sites should be protected based on security needs, this includes
  - Physical security, e.g. strong locks, alarm system, guard service
  - Logical security, e.g. Firewalls, hard drive encryption, Intrusion Detection Systems
  - Organizational security, e.g. clean desk policy

RELEASE PROCESS
PRODUCTION/DELIVERY SECURITY



- Security Measures for HW
  - Security stickers can support integrity protection
  - Unique IDs and tracing of each individual item support detection of item loss or switched items
  - HW labels in design support to identify counterfeit items
- Security Measures for SW or digital items
  - Signatures or Hashing can ensure integrity and authenticity
  - Encryption and access management support confidentiality
- Shipping security
  - Boxes can be additionally protected with secure tape
  - Sealed envelopes can be used
  - Individual parts/information can be shipped separately and/or via different paths or channels

Warning:

Assurance of integrity requires trustworthy developers

It is very hard to detect cheating developers with a high attack potential

SECURITY GUIDANCE OVERVIEW

A security guidance is (a portion of) the user guidance dedicated to security. It should contain the following:

- Preparational Security Guidance
  - Verification of correctness of delivered product
  - Secure installation
  - Requirements for operational environment
- Operational Security Guidance
  - User role concept
  - Description of accessible functions and privileges
  - Secure usage of available interfaces
  - Security parameters
  - Security-relevant events
  - Modes of operation
  - Security measures for the operational environment

PREPARATIONAL SECURITY GUIDANCE VERIFICATION OF DELIVERED PRODUCT

- As for shipping during production, the end user should be able to identify the authenticity of the product
- Some certification schemes like Common Criteria require the user guidance to contain acceptance procedures, which should e.g.
  - ensure that the user has to check that all parts of the product as indicated in the Security Claim have been delivered in the correct version.
  - reflect the steps the user has to perform in order to accept the delivered product that are implied by the developer's delivery procedures.
- Examples for acceptance procedures are
  - get_version command
  - hash of install file
  - name plate/sticker on HW
  - Registration process with product unique ID

PREPARATIONAL SECURITY GUIDANCE SECURE INSTALLATION & ENVIRONMENT

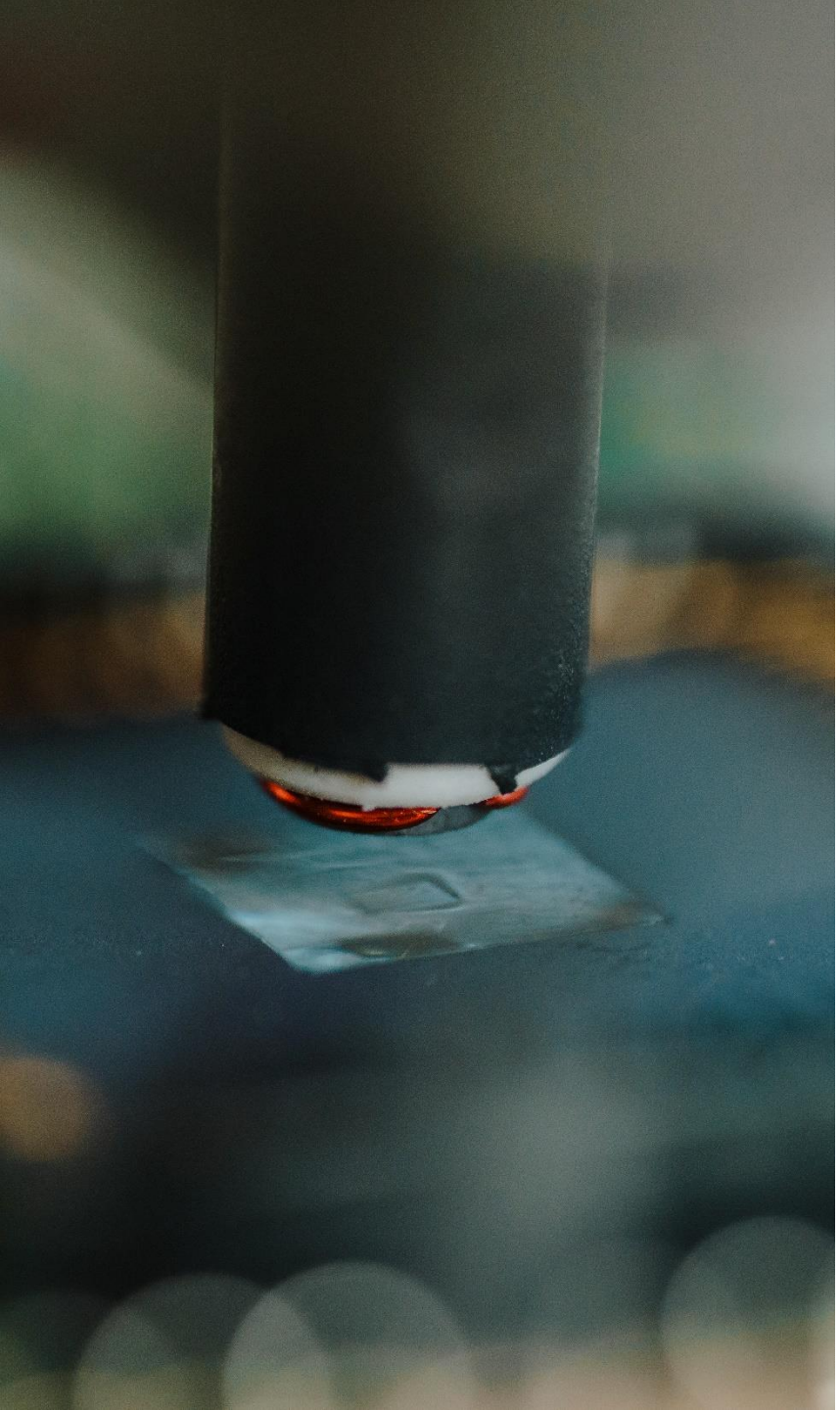
- The user guidance should contain all the steps necessary for secure installation of the TOE
  - If different configurations/settings are possible, the ones regarding security should be clearly identified
  - Such a description can include e.g.
    - for each step a clear scheme for the decision on the next step depending on success, failure or problems at the current step
    - changing the installation specific security characteristics (for example parameters, settings, passwords)
    - handling exceptions and problems
- The user guidance should contain all requirements as well as security measures for the secure operational environment, if applicable (possibly from the security claim)
  - minimum system requirements for secure installation
  - Any other organizational, physical, logical or personnel requirements like
    - Set-up in secured server room
    - Must be operated behind a firewall
    - Password change required

OPERATIONAL SECURITY GUIDANCE
USER ROLE CONCEPT/FUNCTIONS/PRIVILEGES

- Include a user role concept if multiple user roles are supported in OSG
- OSG should describe, for each user role, the user-accessible functions and privileges.
- OSG should contain an overview of the security functionality that is visible at the user interfaces.
- OSG should identify and describe the purpose, behaviour, and interrelationships of the security interfaces and functionality.
- For each user-accessible interface, the user guidance should:
  - describe the method(s) by which the interface is invoked (e.g. command-line, programming-language system call, menu selection, command button);
  - describe the parameters to be set by the user, their particular purposes, valid and default values, and secure and insecure settings of such parameters, both individually or in combination;
  - describe the immediate product response, message, or code returned.

OPERATIONAL SECURITY GUIDANCE EVENTS/MODES OF OPERATION

- OSG should describe, for each user role, each type of security-relevant event that need to be performed, including changes to security settings and operation following failure or operational error.
  - Examples for security-relevant events are
    - audit trail overflow
    - system crash
    - updates to user records, such as when a user account is removed when the user leaves the organization
- OSG should identify all possible modes of operation of the product, e.g.
  - regular mode
  - System recovery mode
  - Super user mode



Secure Updates

AGENDA

- Motivation & Introduction
- Assets, Threats & Attackers
- Security Requirements

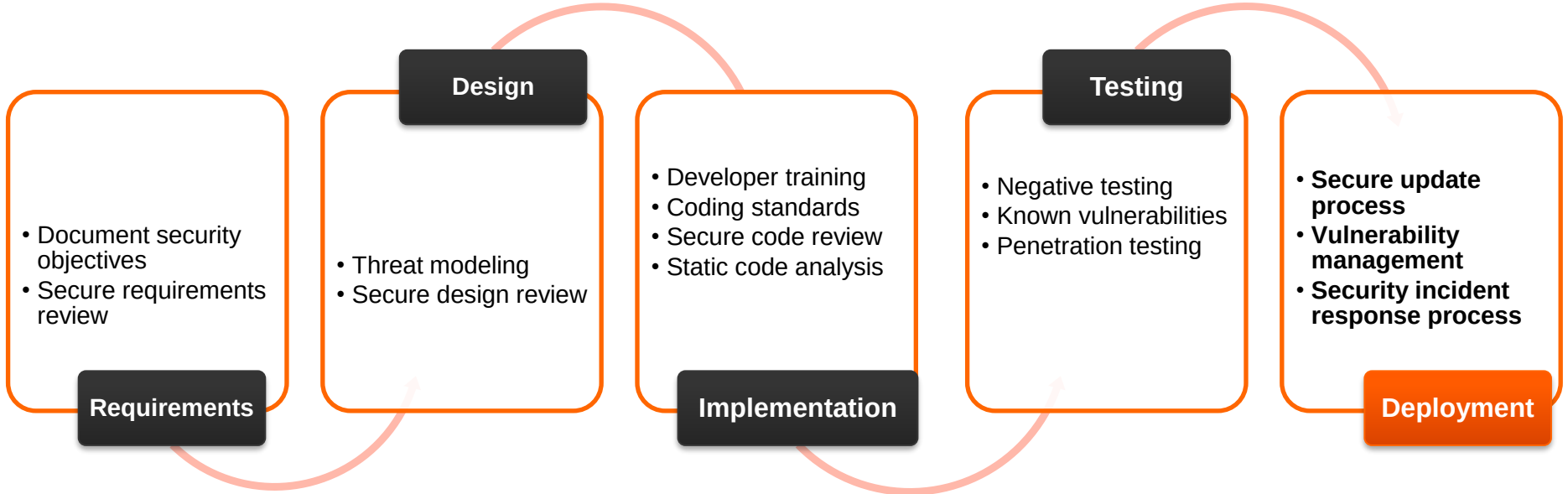
MOTIVATION

- **Security is never static**
  - Newly found vulnerabilities result in new attack vectors
  - Over time, even best security practices at design time can never
    - fully prevent mistakes
    - detect vulnerabilities in already deployed devices

- Vulnerabilities can have serious consequences to
  - Privacy of users
  - Safety of users / citizens
  - Integrity of critical infrastructures

MOTIVATION

- Thus, security must be considered throughout whole product lifecycle & mechanisms must be in place to fix vulnerabilities in the field
- Yet, the field is an untrusted environment
  - Thus, we need a secure in-field update mechanism!



DEVICE UPDATE MECHANISMS

- Bootloader flashes new FW / SW / OS image
- Incremental updates (e.g. via user-space package system)
- Updates of containers (e.g. snap)
- Atomic OS updates (per-file updates; easy to roll-back

ASSETS, THREATS & ATTACKERS

- **Assets**
  - FW / SW image
  - Device integrity (and availability)
- **Threats**
  1. Tampering of FW / SW image
  2. Leakage of FW / SW image (reverse engineering)
  3. Loading of unauthorized FW / SW image
  4. Loading FW / SW image onto unauthorized device
  5. Breaking device through the update process (e.g. via reboot during update)
- **Attackers** are targeting
  - FW / SW image (e.g. while in transport)
  - device during update process
  - backend infrastructure (FW / SW image server, signing server; out of scope)

SECURITY REQUIREMENTS

- **Main security requirements**
  - **Authenticity** of updates
  - **Integrity** of updates
  - **Confidentiality** of updates (during transport & while resting in plain on device)
  - **Integrity** of the device

- **Potential additional security requirements**
  - Long-term security

OTHER REQUIREMENTS

- **Performance**, in terms of
  - Communication effort
  - Computational effort
- **Memory** consumption
- **Energy** efficiency

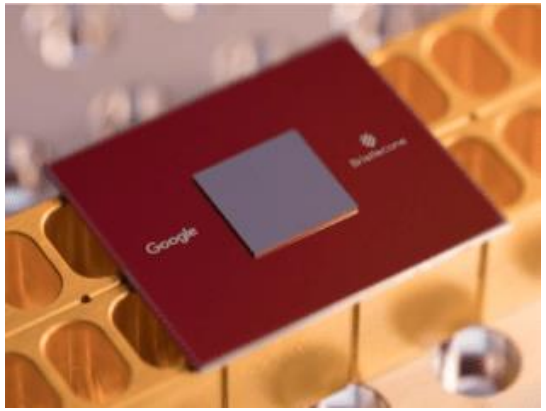
- **Bottom line:** Cryptographic mechanisms must be carefully chosen to fit the requirements!

LONG-TERM SECURITY

- Many devices (e.g. in Automotive or IIoT) are in use for 20+ years
  - It may be necessary to include mechanisms guaranteeing
    - device updateability, and
    - confidentiality of FW / SW updates in the long term
- NIST SP800-57 lists usage periods for concrete security levels
  - e.g. use 256-bit security (AES-256, SHA-512, …) for **2031+**
- Use *perfect forward secrecy* (PFS) e.g. in TLS to protect past image download sessions from long-term key compromise

LONG-TERM SECURITY QUANTUM COMPUTING

- Currently, with the rapid evolution of quantum computers, we are facing an additional challenge:
  - 05/2017: IBM announced 20-qubit QC
  - 11/2017: IBM announced 50-qubits QC
  - 03/2018: Google announced 72-qubits QC
  - 09/2020: IBM expects 1 Mio. qubits by 2030

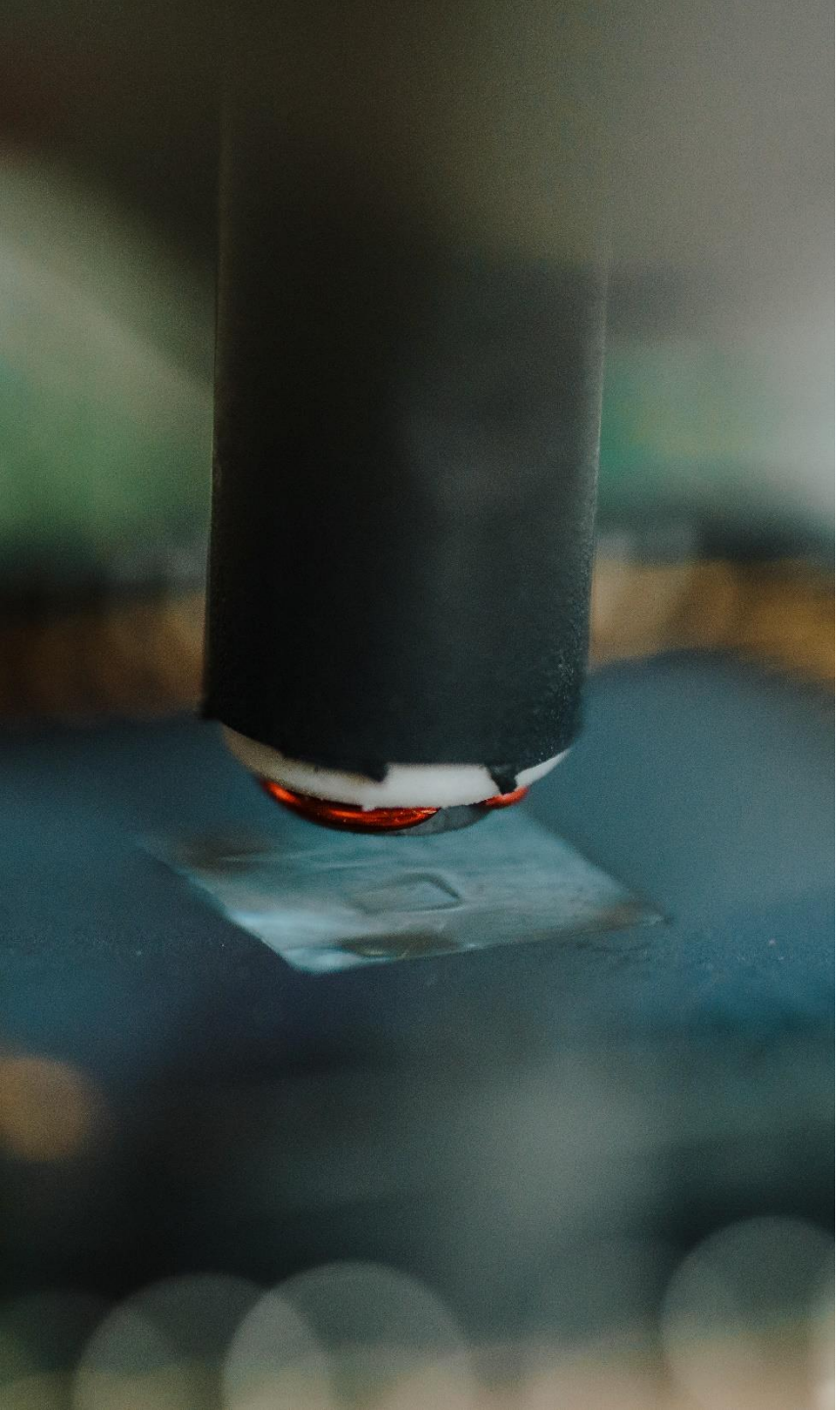


© Google

- We do not know whether in 10 years time classical public-key cryptography (e.g. ECC, RSA) will still be secure
- Depending on use case, it might be necessary to be considered

FINAL THOUGHTS

- Critical security updates should be deployed immediately & automatically, if possible
- More complex systems may require secure updater
  - not only capable of updating the device‘s SW ecosystem, but
  - also able to manage FW updates of sub-components
- This might also require a more heterogenous architecture to handle certificates (e.g. of component manufacturers)



Secure Market Surveillance

WHAT IS MARKET SURVEILLANCE

- EU is regulating market surveillance in the Regulation 2019/1020
- It ensures that products sold in or to Europe comply to the European regulations
- Ensures safety of consumers
- Market surveillance authorities do check products

HOW DOES IT WORK IN REALITY

- Authorized organizations do spot checks of products
- Draw samples at delivery points or in the market
- Confirm that they comply with safety regulations
- Other type of market surveillance can be on behalf of vendors to spot check that products sold under his brand are his products

HOW DOES IT WORK IN REALITY

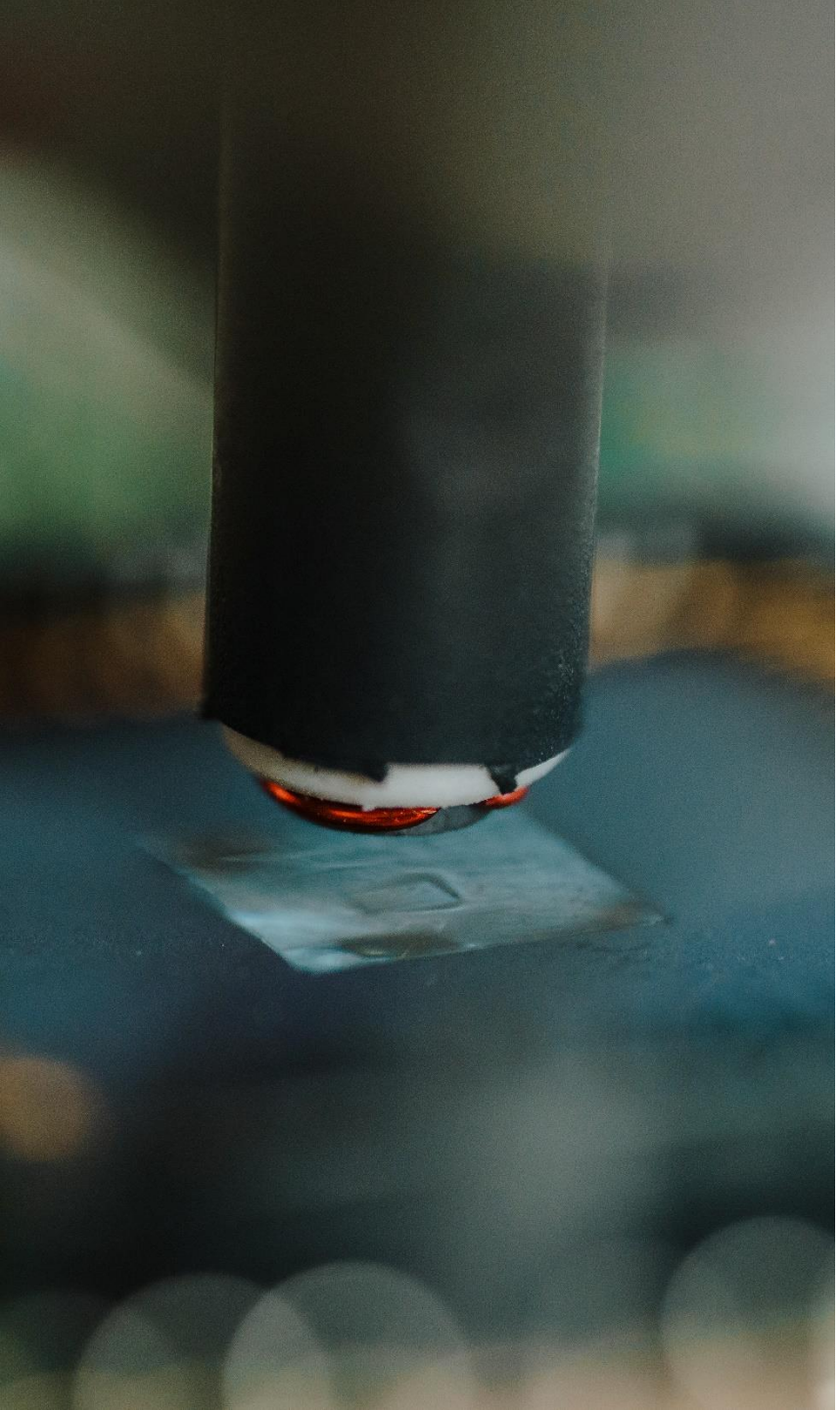
- According to a given process random samples are drawn
- Samples are protocolled
- Samples are tested in lab according to the given regulations
- Report about samples
- Authorities informed about results

WHY IS IT INTERESTING IN THE CONTEXT OF SECURITY

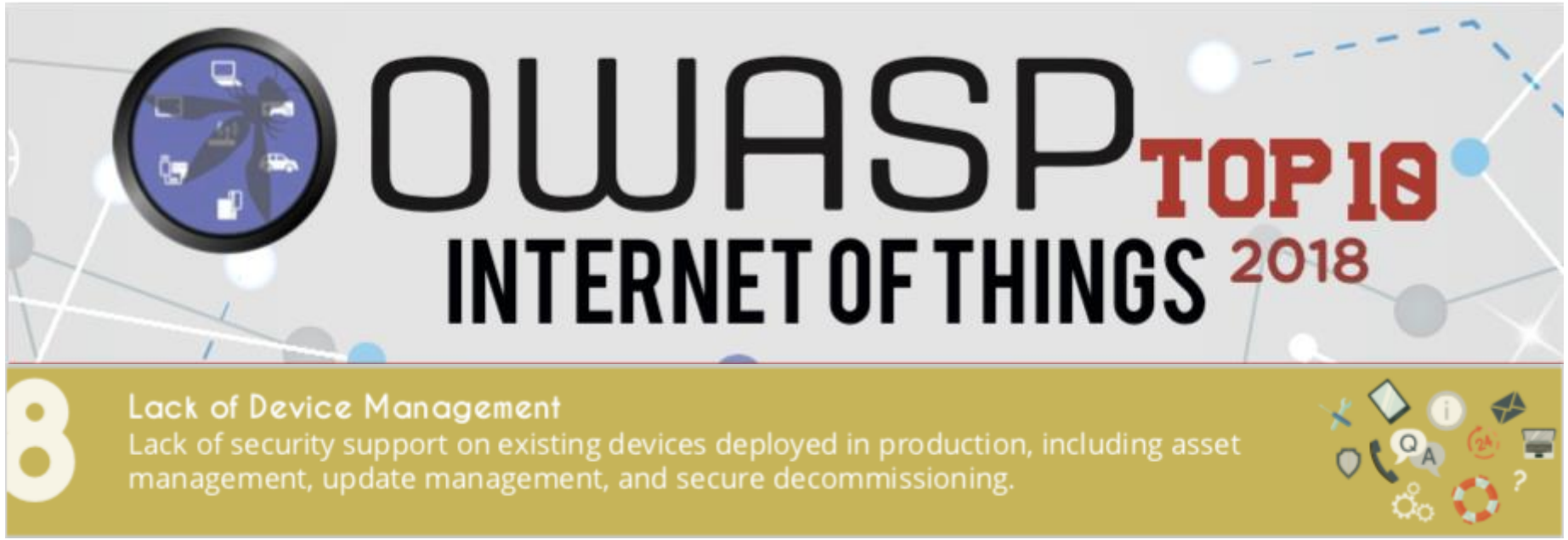
- **Case 1:** Is the product sold, the product which has been checked for security
  - Cheating vendor problem
  - Security can have impact on cost, performance thus sold products could be less secure then tested product
- **Case 2:** Check the configuration of the product
  - Is the product always delivered in a secure configuration
  - Like is the product really using product specific random passwords

WHAT IS THE DIFFICULTY OF MARKET SURVEILLANCE IN REALITY

- Checking the configuration of a product is in most cases simple
- To determine if the product at hand is identical to the product which has been certified is difficult
- There is no means to fingerprint products in an easy way
- How to compare two products if they are identical in terms of HW and SW is an open question



Secure Decommissioning


OWASP TOP 10
INTERNET OF THINGS 2018
8
Lack of Device Management
Lack of security support on existing devices deployed in production, including asset management, update management, and secure decommissioning.

KEY TAKE-AWAYS

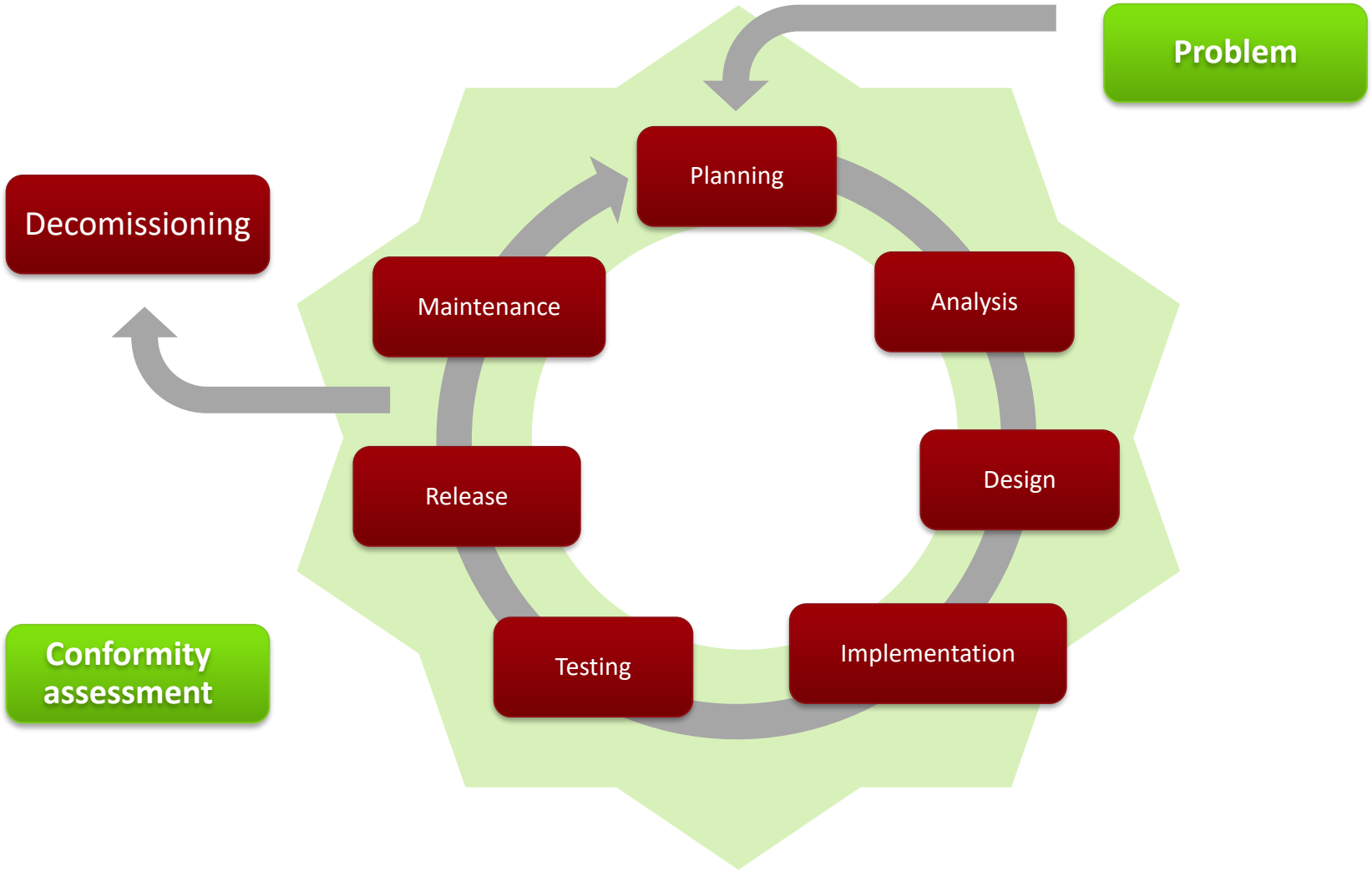
- Device should support deletion of any stored sensitive data when end-of-life is reached

- For products intended to be part of a larger system, decommissioning procedures shall be provided to allow for the product to be removed from the system (and replaced if performing security-critical function) without compromising system security during the product decommissioning process.

EXAMPLES

- Securely wipe all user data
- Securely wipe all secrets
- Securely wipe all configuration
- De-register device from eco system
  - Unauthorize any tokens
  - Revoke any certificates
  - etc.

Secure **Product** Lifecycle



[w] www.yagoba.com, [m] info@yagoba.com
Beethovenstraße 20, 8010 Graz, Austria

Secure Product Lifecycle

Security from Release to Decomissioning

Christoph Herbst

WHEN YOU NEED TO BE SURE

SGS