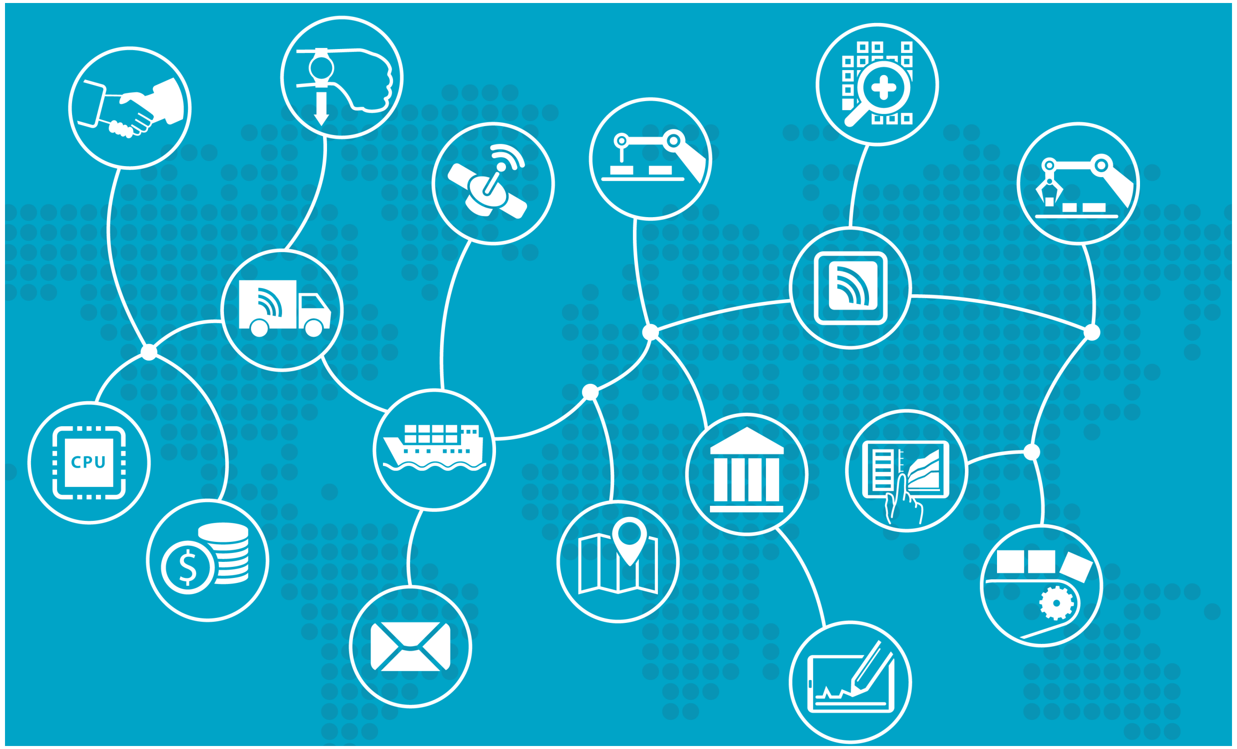
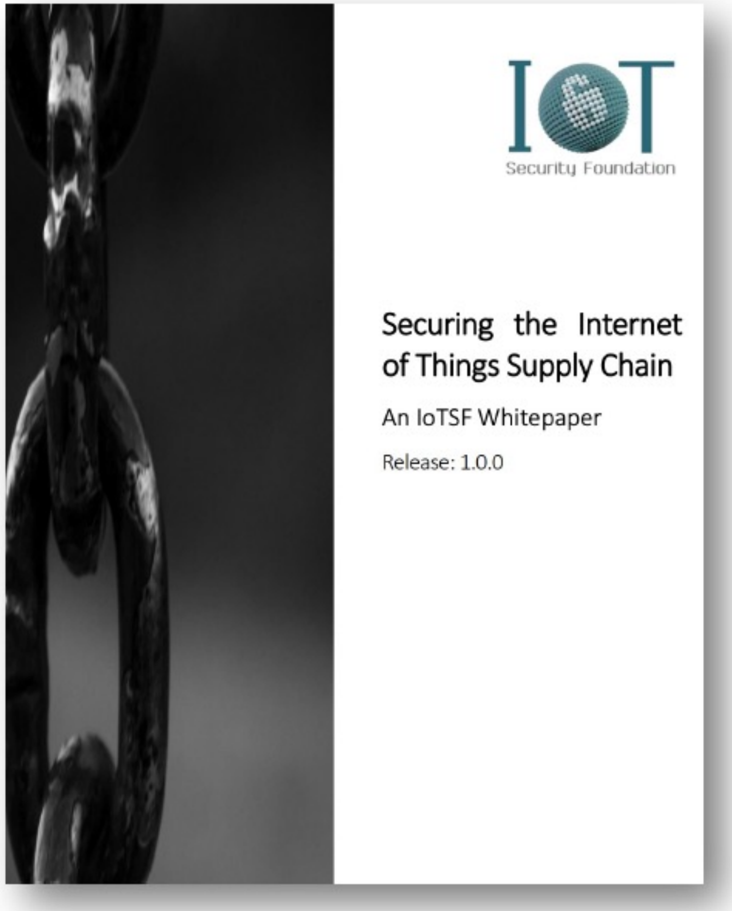
IoT

Security Foundation



Safeguarding IoT Supply Chains

IoT Supply Chain Matters



Securing the Internet of Things Supply Chain

An IoTSF Whitepaper

Release: 1.0.0

Software Bills of Materials for IoT and OT devices

An introduction to the use of SBOMs in the procurement and maintenance of connected devices

An IoTSF Whitepaper

Release: 1.1.0

Ensuring Resilience: Safeguarding IoT Supply Chains

07/27/2024

Ritesh Kumar Kalle, Ph.D

DGM, Cybersecurity R&D

Hitachi India Pvt. Ltd.

Contents

1. Introduction
2. Anatomy of IoT device, Trust and Threats
3. Software Bill of Materials (SBOM)
4. Conclusion

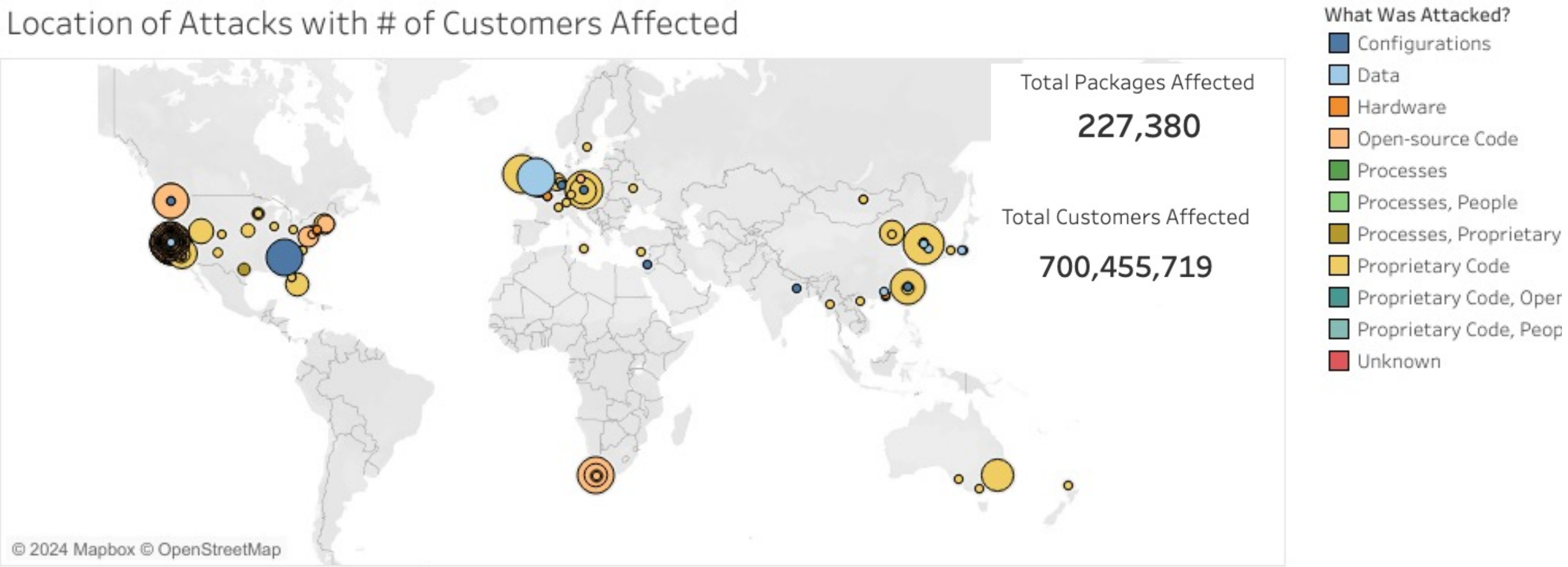
1. Introduction

1-1 IoT device supply chain and attacks

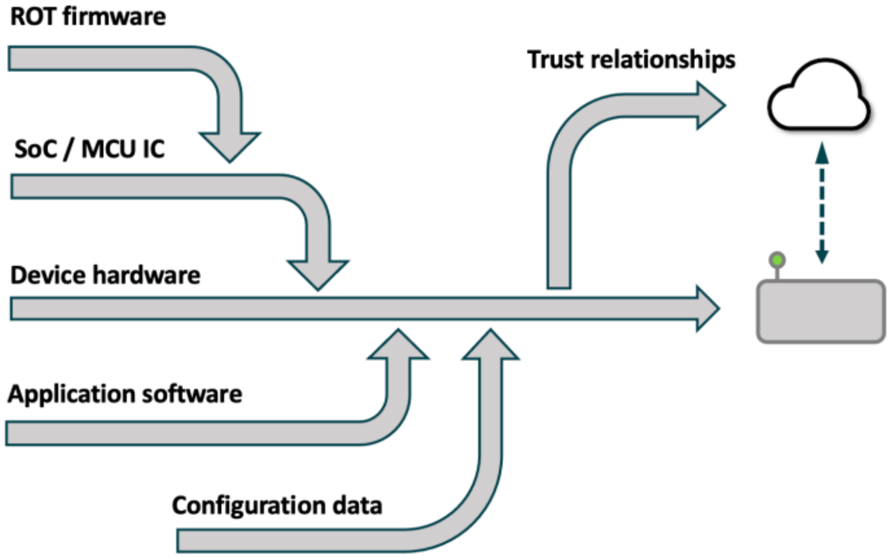
IoT Security Foundation

Neither adherence to design nor procurement practice effective

Attacks are global and diverse



Typical IoT device supply network



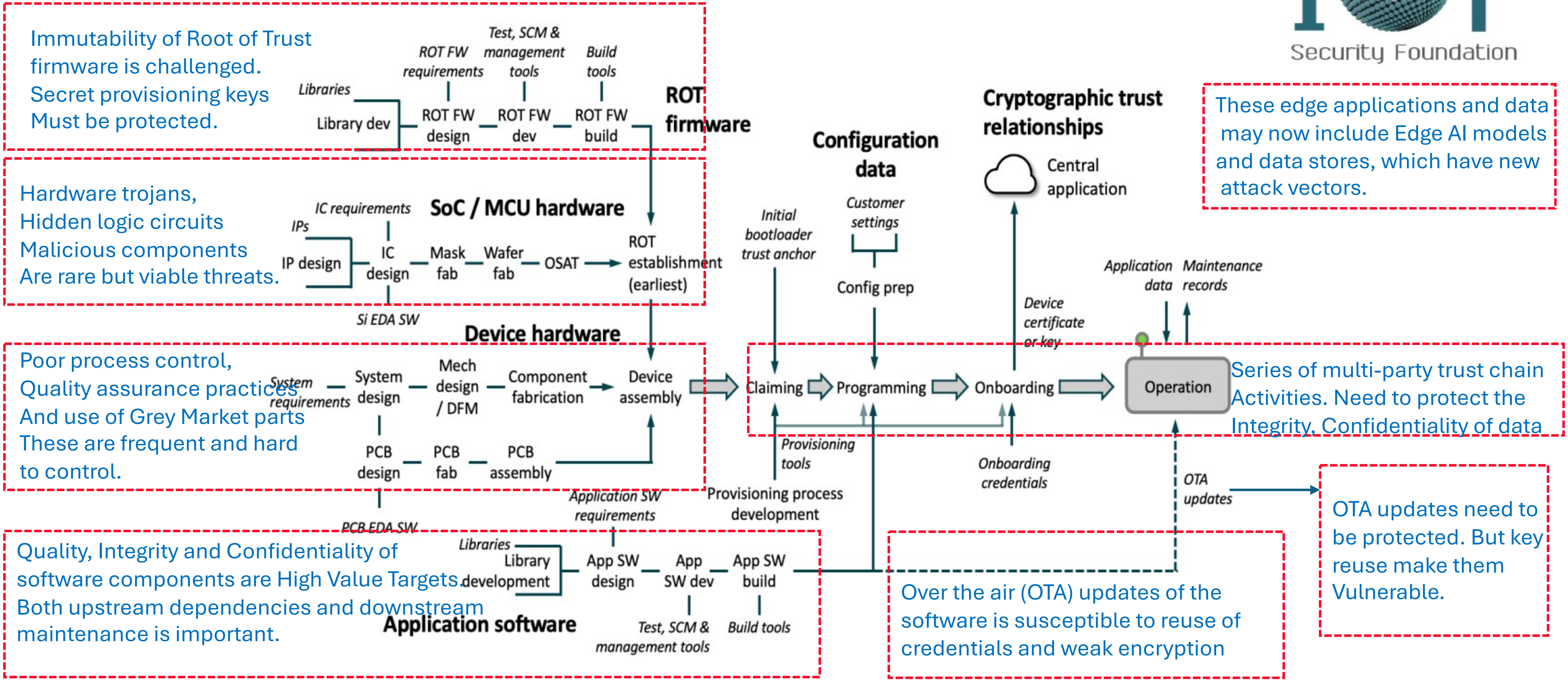
- Supply chain attacks are extremely cost effective from attacker's point of view
- The high “Fan-out” of core components means large customer base is impacted

Ref: [Updated: July 22, 2024] https://www.comparitech.com/software-supply-chain-attacks/

Ref: [IoTSF Whitepaper v1.0.0] Figure 2 Main branches of a typical IoT device supply network

2. Anatomy of IoT device, Trust and Threats

2-1 Anatomy of an IoT device



Immutability of Root of Trust firmware is challenged.
Secret provisioning keys
Must be protected.

Hardware trojans,
Hidden logic circuits
Malicious components
Are rare but viable threats.

Poor process control,
Quality assurance practices
And use of Grey Market parts
These are frequent and hard to control.

Quality, Integrity and Confidentiality of software components are High Value Targets.
Both upstream dependencies and downstream maintenance is important.

These edge applications and data may now include Edge AI models and data stores, which have new attack vectors.

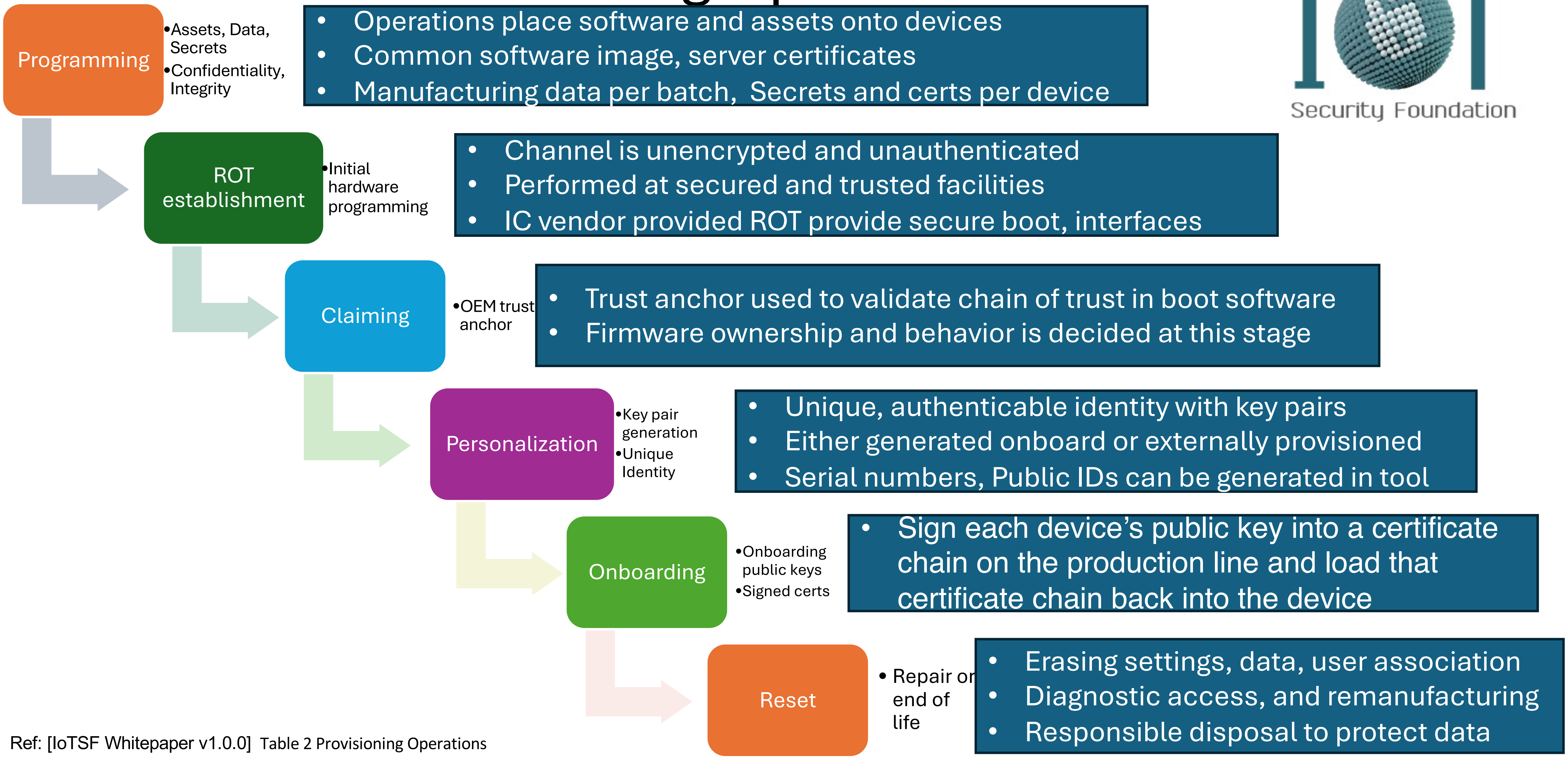
Series of multi-party trust chain Activities. Need to protect the Integrity, Confidentiality of data

OTA updates need to be protected. But key reuse make them Vulnerable.

Over the air (OTA) updates of the software is susceptible to reuse of credentials and weak encryption

Ref: [IoTSF Whitepaper v1.0.0] Figure 3 detailed look at a generalised IoT device supply network

2-2 IoT Device Provisioning Operations



Programming

- Assets, Data, Secrets
- Confidentiality, Integrity

- Operations place software and assets onto devices
- Common software image, server certificates
- Manufacturing data per batch, Secrets and certs per device

ROT establishment

- Initial hardware programming

- Channel is unencrypted and unauthenticated
- Performed at secured and trusted facilities
- IC vendor provided ROT provide secure boot, interfaces

Claiming

- OEM trust anchor

- Trust anchor used to validate chain of trust in boot software
- Firmware ownership and behavior is decided at this stage

Personalization

- Key pair generation
- Unique Identity

- Unique, authenticable identity with key pairs
- Either generated onboard or externally provisioned
- Serial numbers, Public IDs can be generated in tool

Onboarding

- Onboarding public keys
- Signed certs

- Sign each device’s public key into a certificate chain on the production line and load that certificate chain back into the device

Reset

- Repair or end of life

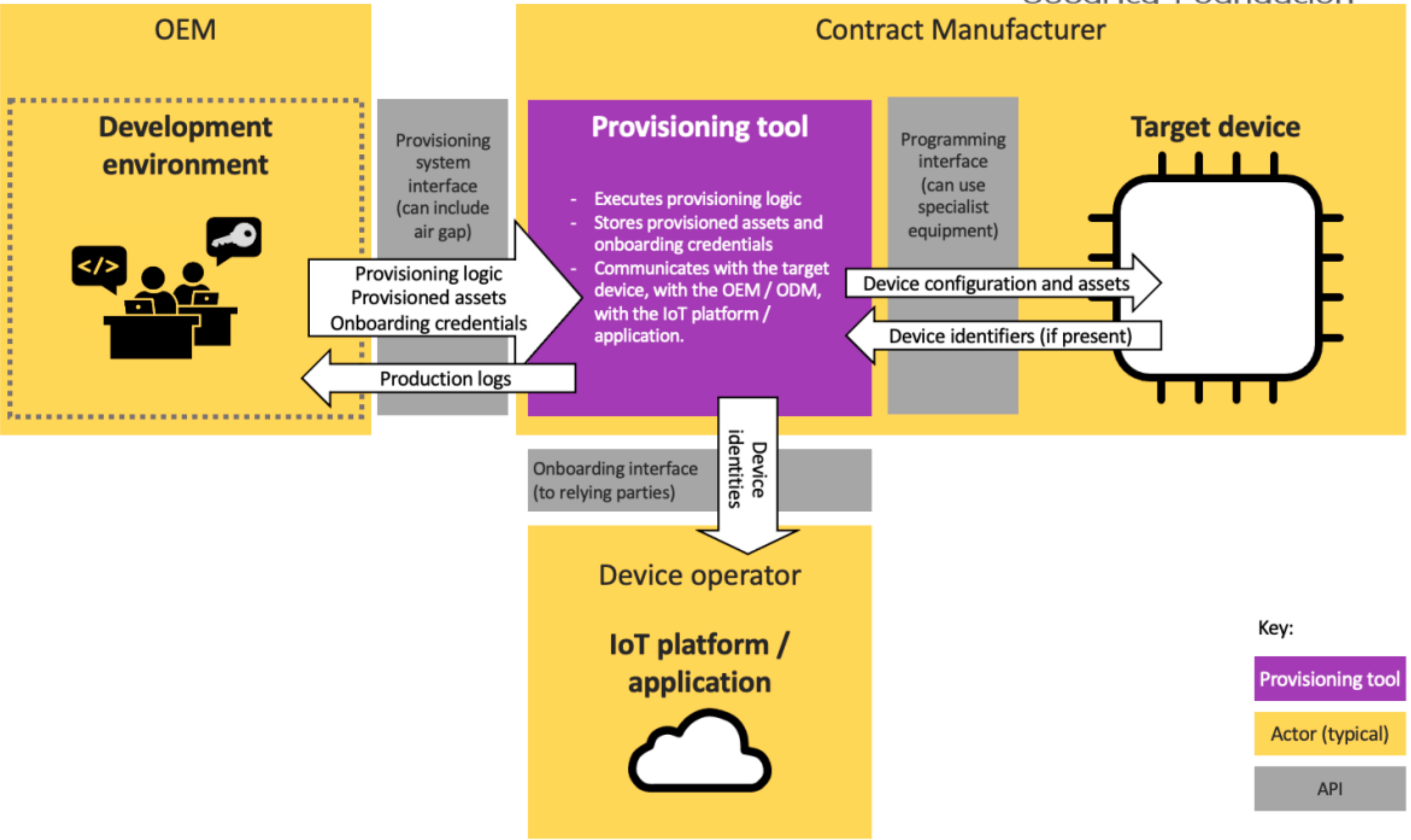
- Erasing settings, data, user association
- Diagnostic access, and remanufacturing
- Responsible disposal to protect data

Ref: [IoTSF Whitepaper v1.0.0] Table 2 Provisioning Operations

2-3 IoT Supply chain trust

What constitutes Trust ?

To deliver a smart device in a *known, functional, and trusted initial state*, its supply chain must *provision it with many software and data assets* and into many trust relationships, often in a sequence of provisioning steps that begins with a blank IC and ends with a fully functional and secured device.



OEM

Development environment

Provisioning system interface (can include air gap)

Provisioning logic
Provisioned assets
Onboarding credentials

Production logs

Contract Manufacturer

Provisioning tool

- Executes provisioning logic
- Stores provisioned assets and onboarding credentials
- Communicates with the target device, with the OEM / ODM, with the IoT platform / application.

Programming interface (can use specialist equipment)

Device configuration and assets

Device identifiers (if present)

Target device

Device identities

Onboarding interface (to relying parties)

Device operator

IoT platform / application

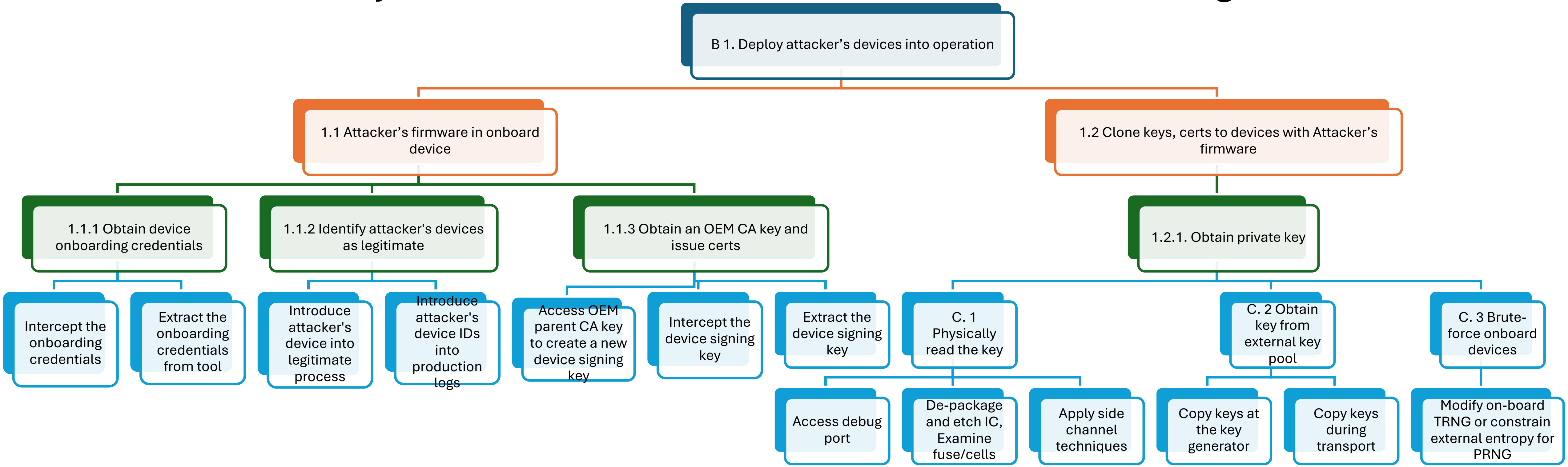
Key:
Provisioning tool
Actor (typical)
API

Ref: [IoTSF Whitepaper v1.0.0] Figure 4 Generic provisioning operation

2-4 Threat Model – Attack Trees

Example - Disrupt or monitor operators of devices

- To defend against attacks, Disrupt or weaken the the chain of conditions
- Refer to IoT Security assurance framework recommendations for mitigations

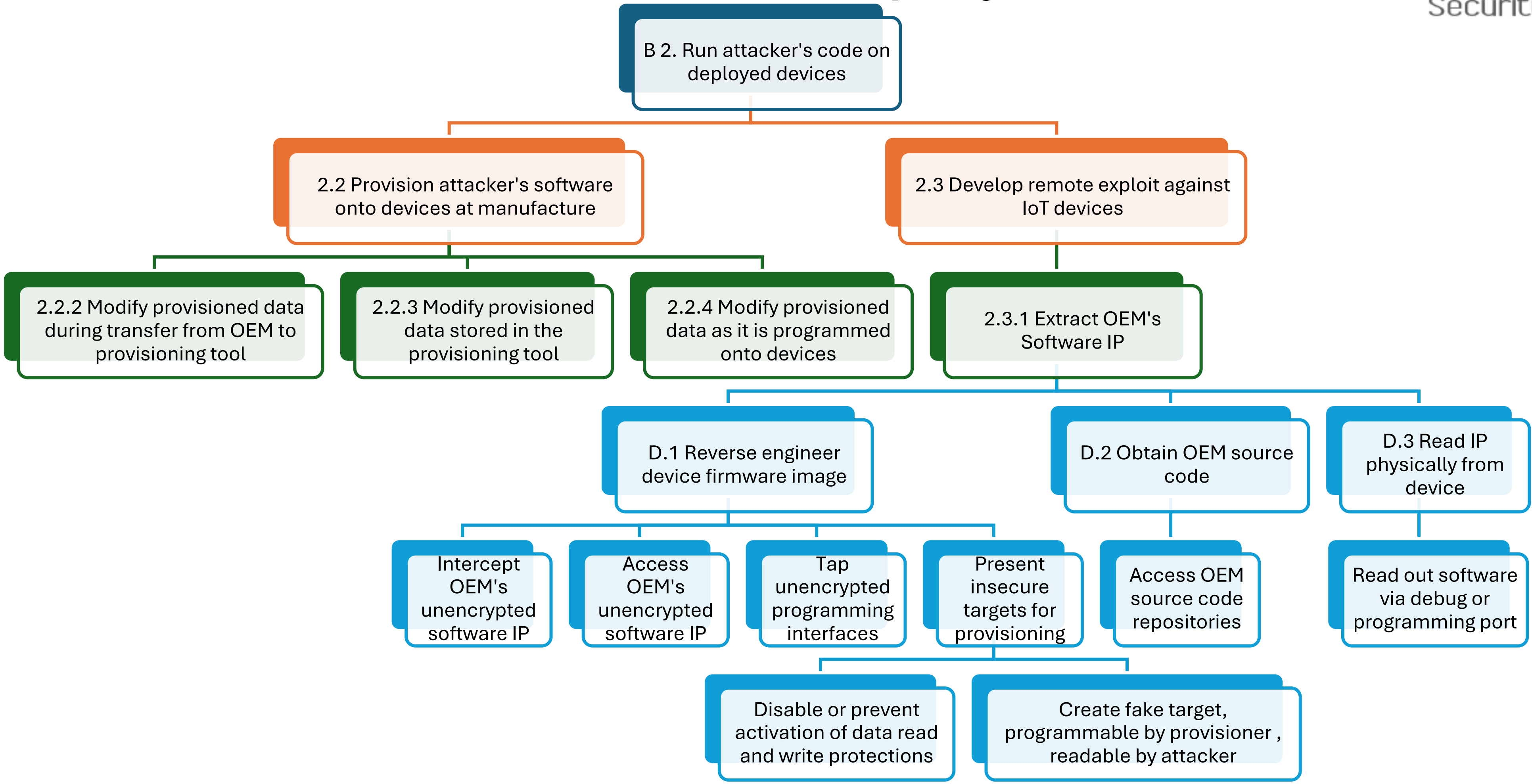


- B 1. Deploy attacker’s devices into operation
  - 1.1 Attacker’s firmware in onboard device
    - 1.1.1 Obtain device onboarding credentials
      - Intercept the onboarding credentials
      - Extract the onboarding credentials from tool
    - 1.1.2 Identify attacker's devices as legitimate
      - Introduce attacker's device into legitimate process
      - Introduce attacker's device IDs into production logs
    - 1.1.3 Obtain an OEM CA key and issue certs
      - Access OEM parent CA key to create a new device signing key
      - Intercept the device signing key
      - Extract the device signing key
  - 1.2 Clone keys, certs to devices with Attacker’s firmware
    - 1.2.1. Obtain private key
      - C. 1 Physically read the key
        - Access debug port
        - De-package and etch IC, Examine fuse/cells
        - Apply side channel techniques
      - C. 2 Obtain key from external key pool
        - Copy keys at the key generator
        - Copy keys during transport
      - C. 3 Brute-force onboard devices
        - Modify on-board TRNG or constrain external entropy for PRNG

Ref: [IoTSF Whitepaper v1.0.0] Appendix B Attack Tree

2-5 Threat Model – Attack Trees

Example - Run attacker’s code on deployed devices



B 2. Run attacker's code on deployed devices

2.2 Provision attacker's software onto devices at manufacture

2.3 Develop remote exploit against IoT devices

2.2.2 Modify provisioned data during transfer from OEM to provisioning tool

2.2.3 Modify provisioned data stored in the provisioning tool

2.2.4 Modify provisioned data as it is programmed onto devices

2.3.1 Extract OEM's Software IP

D.1 Reverse engineer device firmware image

D.2 Obtain OEM source code

D.3 Read IP physically from device

Intercept OEM's unencrypted software IP

Access OEM's unencrypted software IP

Tap unencrypted programming interfaces

Present insecure targets for provisioning

Access OEM source code repositories

Read out software via debug or programming port

Disable or prevent activation of data read and write protections

Create fake target, programmable by provisioner , readable by attacker

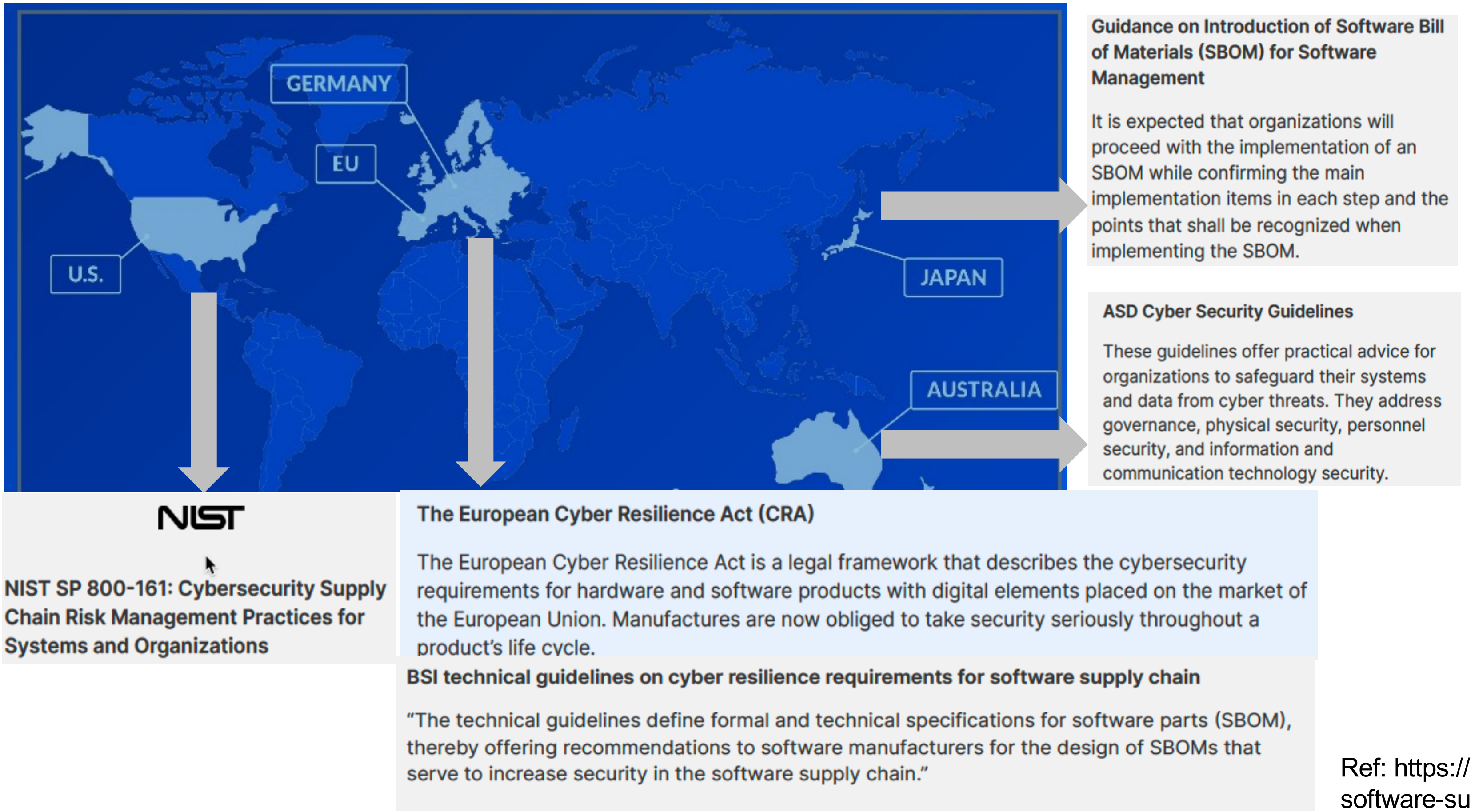
Ref: [IoTSF Whitepaper v1.0.0] Appendix B Attack Tree

3. Software Bill of Materials (SBOM)

3-1 Regulations driving adoption of SBOM

Software supply chain security regulations expanding

SBOMs are enablers for transparency



Guidance on Introduction of Software Bill of Materials (SBOM) for Software Management

It is expected that organizations will proceed with the implementation of an SBOM while confirming the main implementation items in each step and the points that shall be recognized when implementing the SBOM.

ASD Cyber Security Guidelines

These guidelines offer practical advice for organizations to safeguard their systems and data from cyber threats. They address governance, physical security, personnel security, and information and communication technology security.

NIST SP 800-161: Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations

The European Cyber Resilience Act (CRA)

The European Cyber Resilience Act is a legal framework that describes the cybersecurity requirements for hardware and software products with digital elements placed on the market of the European Union. Manufactures are now obliged to take security seriously throughout a product's life cycle.

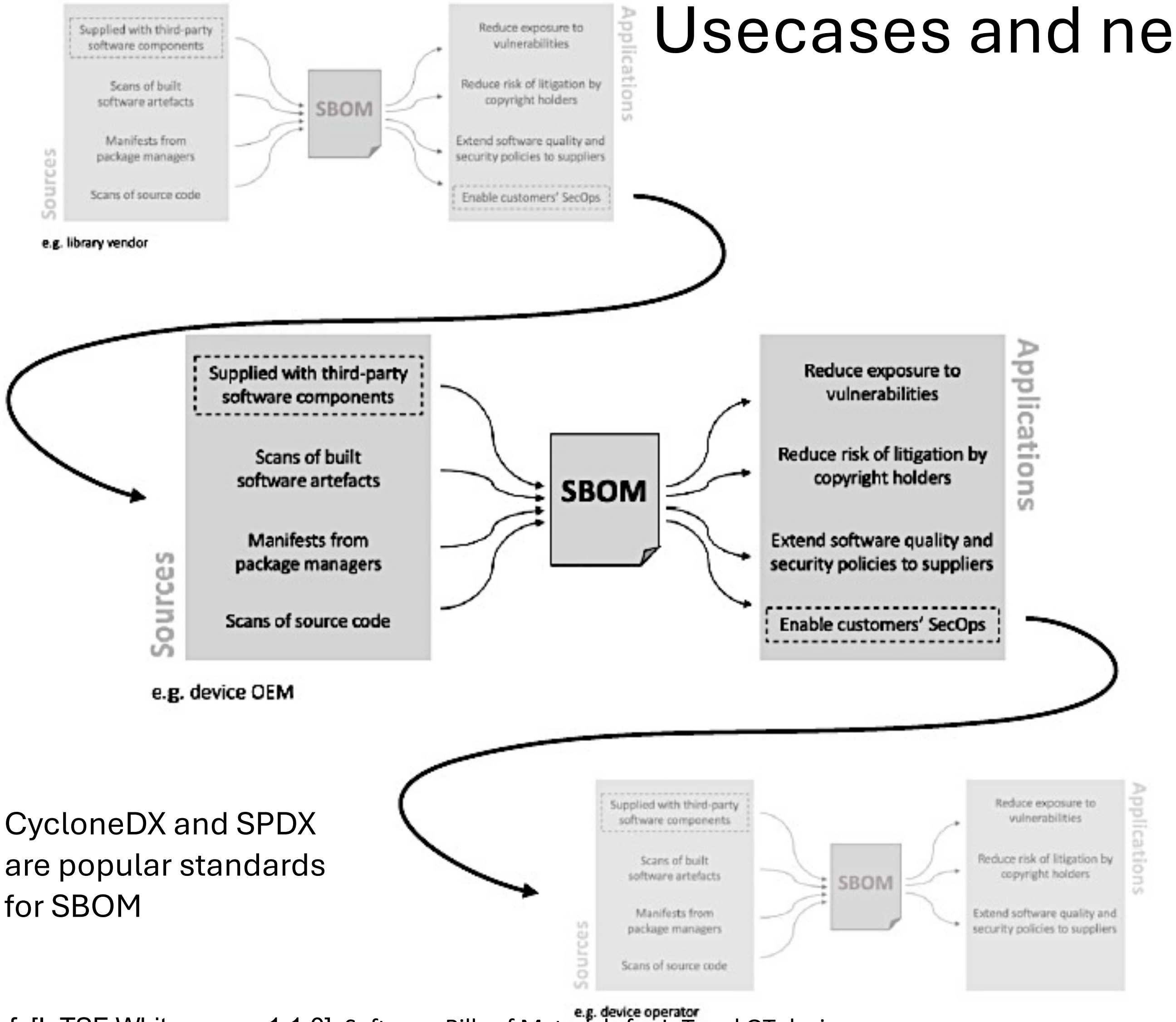
BSI technical guidelines on cyber resilience requirements for software supply chain

"The technical guidelines define formal and technical specifications for software parts (SBOM), thereby offering recommendations to software manufacturers for the design of SBOMs that serve to increase security in the software supply chain."

Ref: https://scribesecurity.com/resource/ensuring-the-security-of-software-supply-chains-meeting-compliance-and-legal-obligations/

3-2 Sharing SBOMs down the supply chain

Usecases and need for sharing of SBOM



Supplied with third-party software components

Scans of built software artefacts

Manifests from package managers

Scans of source code

Sources

SBOM

Reduce exposure to vulnerabilities

Reduce risk of litigation by copyright holders

Extend software quality and security policies to suppliers

Enable customers' SecOps

Applications

e.g. library vendor

Supplied with third-party software components

Scans of built software artefacts

Manifests from package managers

Scans of source code

Sources

SBOM

Reduce exposure to vulnerabilities

Reduce risk of litigation by copyright holders

Extend software quality and security policies to suppliers

Enable customers' SecOps

Applications

e.g. device OEM

Supplied with third-party software components

Scans of built software artefacts

Manifests from package managers

Scans of source code

Sources

SBOM

Reduce exposure to vulnerabilities

Reduce risk of litigation by copyright holders

Extend software quality and security policies to suppliers

Applications

e.g. device operator

CycloneDX and SPDX are popular standards for SBOM

Component type	Recommended methods of providing SBOM documentation
Source code libraries	SBOM is included in top-level directory
Binaries (for linking into downstream projects)	SBOM is included in an archive with the binary
Packaged binaries (to be run in OS environments)	SBOM is included in the package SBOM is posted as a web resource and its URL included in the package metadata
Device images (for installation on IoT/OT devices by operators)	SBOM is included in an archive with the device image
Device images (installed by manufacturer during production or via remote update mechanism)	SBOM is posted as a web resource AND Where devices connect to a central management service: Devices report to their central management service the URL of the SBOM Devices report to their central management service their software version number. Manufacturer publishes a web page or resource listing SBOM URLs against software version numbers for each model of device [6]. Where no central management service is used: Devices serve the SBOM URL at .well-known/sbom [rfc-8615] [draft-ietf-opsawg-sbom-access-13] Devices serve the SBOM via an extended Manufacturer Usage Description (MUD) [rfc-8520] [draft-lear-opsawg-mud-sbom-00]

Ref: [IoTSF Whitepaper v1.1.0] Software Bills of Materials for IoT and OT devices

4. Conclusion

- ❑ Safeguarding IoT supply chain is a shared responsibility of all stakeholders in the Industrial IoT ecosystem.
- ❑ Transparency and accountability can be enabled through adoption of open standards.
- ❑ A chain is only as strong as its weakest link!

END

Ensuring Resilience: Safeguarding IoT Supply Chains

07/27/2024

Ritesh Kalle, Ph.D

DGM, Cybersecurity R&D
Hitachi India Pvt. Ltd.