About the Bootstrapping of Security in IoT

Bootstrapping is the act of adding an IoT component to a site e.g. a smart home or an automation facility. This stage in the lifecycle of an IoT component happens after its manufacturing and before operation. A single IoT component may pass the bootstrapping stage multiply during its lifecycle. Figure 1 provides a model for the lifecycle of an IoT component:

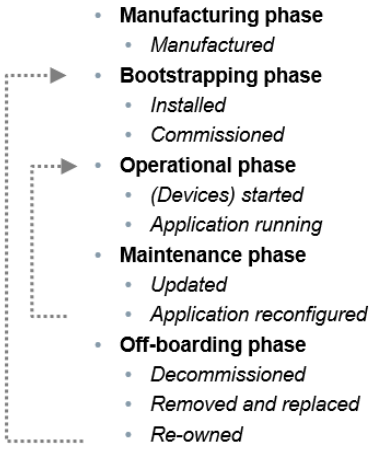

Figure 1: Lifecycle of IoT Components

The bootstrapping stage usually comprises an installation step for the physical setup (e.g. mounting, connecting power and/or network cables) and a commissioning step for the logical setup (e.g. configuring) of the IoT component.

With respect to security, the bootstrapping stage is one of the most critical stages in the lifecycle of an IoT component: during this stage the initial keys resp. credentials that facilitate (secure) interactions between the IoT component and other components (thing-to-thing) and/or services (thing-to-service or service-to-thing) in the site are determined.

The Challenge

During the bootstrapping phase the IoT component (as joiner or pledge) and the site (as target) need to establish mutual trust as well as keying associations – if security is an objective for the considered IoT deployment. This comprises following tasks:

a) The site needs to authenticate and authorize the IoT component: *What is this component? Do I want it? How to communicate securely with it?*

b) The IoT component needs to authenticate and authorize the site: *What is this site? Should I join it? How to communicate securely with it?*

Figure 2 illustrates these two core tasks during bootstrapping:

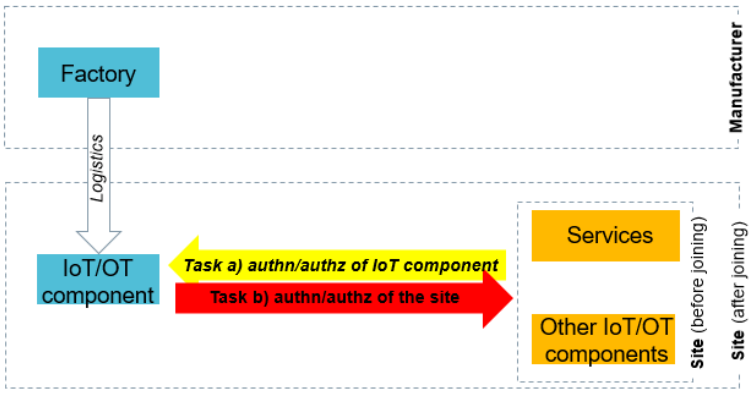

Figure 2: Core Tasks During Bootstrapping

Security requirements apply for the realization of the bootstrapping tasks a) and b). This has an obvious reason: insecure practices will result in weaknesses that may lead to security breaches.

Efficiency requirements also apply. It shall be feasible to automate the bootstrapping tasks a) and b) in order to support sites with large numbers of IoT components and/or users resp. staff who are no security experts.

Common Practices

The commonly encountered practices for the two core bootstrapping tasks a) and b) tend to compromise security for the benefit of ease-of-use or vice versa. Solutions that achieve ease-of-use and security are hard to find as of today. This chapter gives an overview of current practices that are common.

Ease-of-Use

Among the IoT solutions that prefer ease-of-use over security, various approaches including the following ones can be found:

1. No security[1] at all
2. Imprinting of site secrets/keys/credentials during manufacturing; the same value is shared among product instances
3. Imprinting of site secrets/keys/credentials during manufacturing; utilizing product instance-specific values

These approaches are briefly assessed as follows:

1. Not supporting/using security at all avoids complexity regarding the bootstrapping tasks a) and b) but is limited to naïve security models in style of "*accept any IoT component*" resp. "*accept any site*". Such naïve models present a mismatch for many IoT deployments.
2. IoT components that are shipped with default username/password-credentials are an important example of this approach. Obviously, this is critical from a security point-of-view. For instance, the Mirai botnet exploited IoT components that were shipped with default username/password-credentials (shared across product instances) to perform a massive DDoS attack against IT services on the Internet. See [11] for more information about Mirai.

[1] Means that make use of secrets, keys and/or credentials

3. This approach is frequently encountered with devices such as WLAN/Wi-Fi access points (case of WPA security). It is limited with respect to security (secret information shared with a 3rd party – the manufacturer). It also presents challenges with respect to the production process (product is not fully generic). Moreover, this approach depends on some (manual) synchronization between the IoT component and the site.

Security

Among the IoT solutions that prefer security over ease-of-use, several approaches including the following ones can be found:

4. IoT component are shipped with no secrets/keys/credentials assigned during the manufacturing process
5. IoT components are shipped with secrets/keys/credentials assigned during the manufacturing process

These approaches are also briefly assessed as follows:

4. This approach is as known as “resurrecting duckling” model and is described in [10]. It depends on (manual) credentialing of the IoT component as well as synchronization between the IoT component and the site. Moreover, the reset operation is unprotected (unless specific measures are taken).
5. This approach was kicked-off be IEEE 802.1 AR (see [1]) which focused on asymmetric keys/credentials that are assigned by the manufacturer and shipped with the product. IEEE 802.1 AR describes the automated supply of asymmetric site keys/credentials to this product. This relies on the principle of cryptographic separation. But no protocol is specified by IEEE 802.1 AR for doing the site-by-manufacturer credentialing trick. EST (RFC 7030, see [8]) is an Internet standard that fills this gap by defining a blend of HTTP and TLS exchanges for this trick. In combination with information about allowances this automates the bootstrapping task a) as well as the supply of site-specific EE credentials/certificates to IoT components. But EST depends on manual steps to acquire site-specific CA certificate objects as an IoT component.

Not Yet Championed Steps

The prior art (e.g. IEEE 802.1 AR, IETF EST) provides a recipe for automating task a):

- As manufacturer: ship IoT components with manufacturer credentials (IDevID)
- As site: use these credentials to authenticate the component. In addition, use information about allowances to authorize the component.

This also provides a recipe for automating a subset of task b)

- As site resp. IoT component: based on authenticating and authorizing the IoT component assign a site EE credential (LDevID) to the IoT component.

But the prior art does not provide a recipe for automating the remainders of task b): the actual authentication and authorization of the site by the joining IoT component. This presents the hard part for the bootstrapping of security in IoT.

Since initial secrets/keys/credentials on the highest level of keying hierarchy cannot be protected by cryptographic techniques, it is a common practice in IT-security to rely on

out-of-band channels and/or manual tasks for supplying initial secrets/keys/credentials in a distributed system. These practices do not match the needs of IoT well.

Relevant Initiatives

Solutions that combine ease-of-use and security for the bootstrapping of IoT components are difficult to find as of now: new mechanisms are required to achieve secure and zero-touch credentialing to a full extent. They did not exist before 2018 and emerge from current work - for instance the Anima (Autonomic Networking Integrated Model and Approach) initiative at the IETF. This chapter investigates the IETF Anima as well as selected other IETF initiatives that are aiming to fill the identified gap.

Anima

IETF Anima (see [2] and [9]) addresses the hard part b) by using an indirection: a site-facing manufacturer service is used to authenticate and authorize the site during IoT component bootstrapping. Figure 3 illustrates this solution approach for part b)

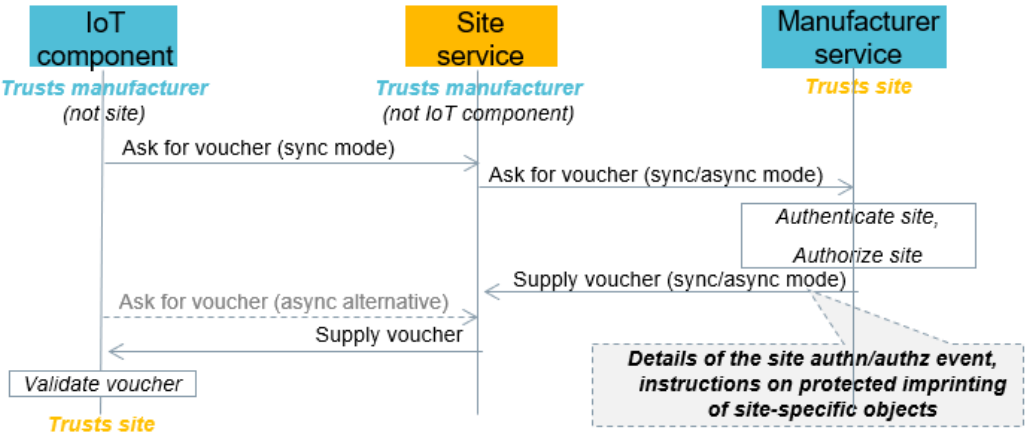

Figure 3: IETF Anima Approach for Part B)

This exchange happens during the bootstrapping of an IoT component in a site. It uses a manufacturer signed data object called voucher by which the manufacturer informs the IoT component about the event of site authentication and authorization. This object also facilitates the imprinting of site-specific information esp. the site CA certificate. The supply of the voucher object to the IoT component is enabled by a site service (called registrar or domain registrar). The acquisition of the voucher can happen in synchronous as well as asynchronous modes (nonce-less vouchers). The IoT component is meant to behave as instructed by the contents of the voucher object.

IETF Anima defines exchanges for voucher acquisition (and hence the imprinting of site CA certificates) by means of HTTP-over-TLS and JSON. Anima uses EST for the complementary acquisition of site EE certificates (see [8]). An accompanying specification translates this to CoAP-over-DTLS and CBOR (see [5] and [6]).

In addition to supplying security objects during bootstrapping, IETF Anima also considers aspects such as supplying (local) network connectivity and discovery of services/components.

6tisch Zero Touch

IETF 6tisch Zero Touch (see [4]) largely resembles the secure and zero-touch bootstrapping approach of IETF Anima – with some deviations on a level of detail that is not elaborated herein. On the considered level, 6tisch Zero Touch uses the same system architecture, exchanges and objects for the purpose of bootstrapping incl. vouchers as manufacturer-issued endorsement objects.

6tisch Minimal Security

IETF 6tisch Minimal Security (see [3]) does not require manufacturer services. This initiative does not deal with manufacturer-issued endorsements (aka vouchers). It does also not deliver a zero-touch approach: it requires the manual establishment of an initial site key in form of a shared secret between the IoT component and a site service (registrar aka domain registrar).

Netconf SZTP

IETF Netconf SZTP (see [7]) does not require site services. This initiative specifies IoT component-facing manufacturer services. This implies a subtle difference in the notion of “bootstrapping” since Netconf is unaware of a site as an ensemble of IoT components and services that jointly fulfill a purpose. However, Netconf also uses voucher objects based on RFC 8366 (see [9], called ownership voucher in Netconf).

Assessment

The investigated IETF initiatives for bootstrapping rely on site and/or manufacturer services:

- Site and manufacturer services (site-facing): Anima, 6tisch Zero Touch
- Site services: 6tisch Minimal Security
- Manufacturer services (component-facing): Netconf

Site-facing manufacturer services are an interesting concept since they allow manufacturers to engage in CRM use cases. However, the current IETF Anima specifications come with some limitations including:

- Brown-field friendliness: the current addressing scheme is no good fit for manufacturers with many, small pools of unique product serial numbers
- Sustainability: service API versioning is not yet covered; message objects are self-contained but do not always embody meta-information about their structure
- Scalability: bulk operation modes are not yet supported
- Unified credentialing: equipping a dedicated IoT component with multiple, site-specific credentials (LDevIDs) that are bound to dedicated application domains can be accommodated but is not yet elaborated

Summary

Secure and zero-touch bootstrapping is a key concern in IoT security. It is not yet championed to a full extent. Important innovations are happening right now, on international level e.g. the IETF Anima working group.

Since blueprints for the solution of secure and zero-touch bootstrapping emerge now, one can expect to obtain core specification elements today. Fully blown specifications covering all possible aspects and/or adoptions to target domains are still to come.

Some of the emerging solutions allow to do more than secure and zero-touch bootstrapping: their security architecture provides components and exchanges that empower IoT component manufacturers to engage in CRM use cases.

References

[1] IEEE 802.1AR-2009, *IEEE Standard for Local and Metropolitan Area Networks – Secure Device Identity*, 2009

[2] IETF BRSKI: *Bootstrapping Remote Secure Key Infrastructures (BRSKI)*, Draft (work-in-progress), 2019

[3] IETF 6tisch Minimal Security: *Minimal Security Framework for 6TiSCH, Draft (work-in-progress), 2019*

[4] IETF 6tisch Zero-Touch: *6tisch Zero-Touch Secure Join protocol, Draft (work-in-progress), 2018 (expired)*

[5] IETF Constrained Voucher: *Constrained Voucher Artifacts for Bootstrapping Protocols,* Draft (work-in-progress), 2019

[6] IETF EST-coaps: *EST over secure CoAP (EST-coaps),* Draft (work-in-progress), 2019

[7] IETF Netconf SZTP: *Secure Zero Touch Provisioning (SZTP),* Draft (work-in-progress), 2019

[8] IETF RFC 7030: *Enrollment over Secure Transport*, RFC 7030, 2013

[9] IETF RFC 8366: *A Voucher Artifact for Bootstrapping Protocols*, RFC 8366, 2018

[10] Stajano, F.; Anderson, R.: *The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks.* In: Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Computer Science, 1999

[11] Wikipedia: *Mirai (malware)*. Retrieved May 25, 2019

Author

Oliver Pfaff

Siemens AG, CT RDA ITS

Mail: oliver.pfaff@siemens.com