RFC 8572

Secure Zero Touch Provisioning (SZTP)

(RFC Published April 2019)

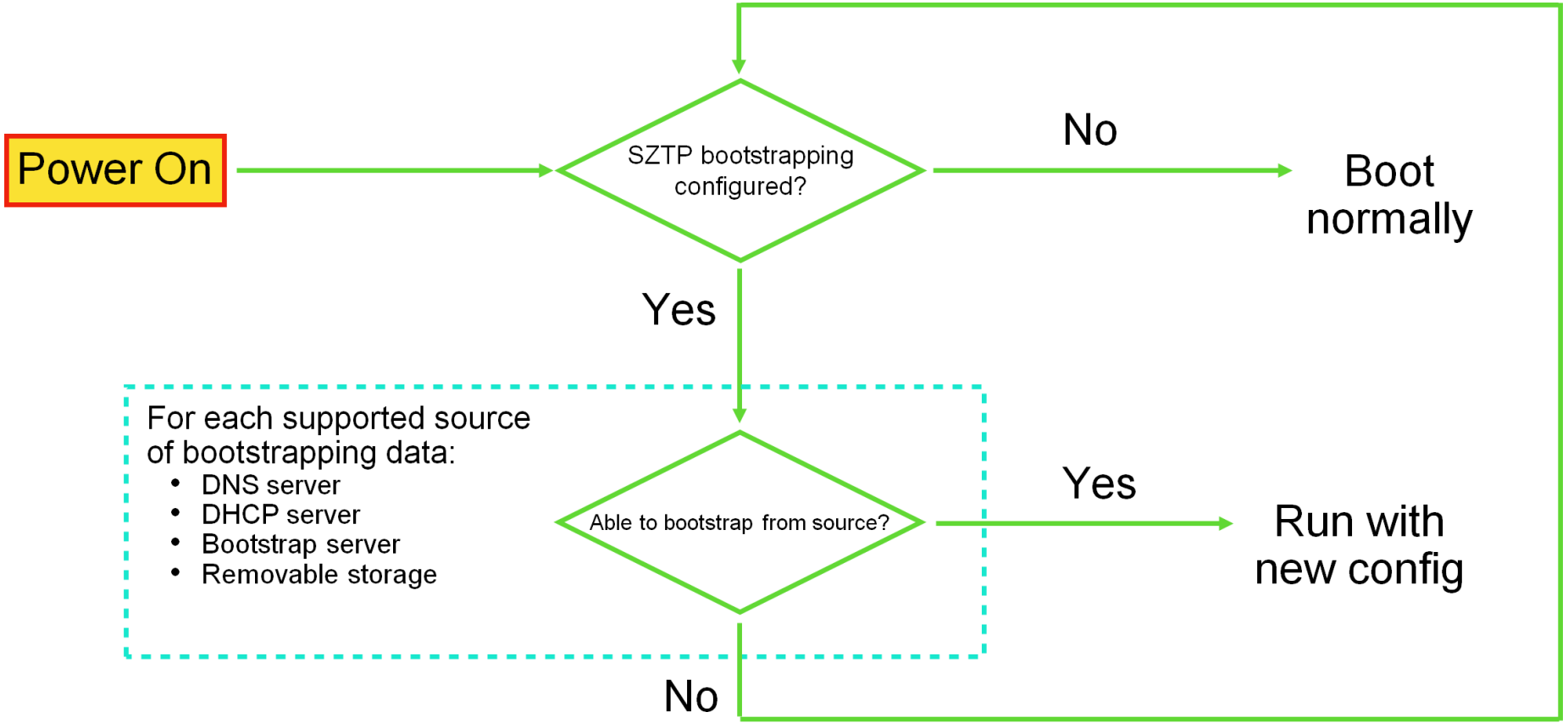
RFC 8572 / SZTP

- Presents a technique to securely provision a networking device when it is booting in a factory-default state.
- Variations in the solution enable it to be used on both public and private networks.
- The provisioning steps are able to update the boot image, commit an initial configuration, and execute arbitrary scripts to address auxiliary needs.
- The updated device is subsequently able to establish secure connections with other systems.

Key Characteristics

- Protocol is device-initiated (on boot, whenever device in a factory default state)
- Supports both Internet and non-Internet based deployments.
- Several possible sources of bootstrapping data:
  - Removable storage device, DHCP server, DNS server, SZTP Bootstrap server, etc.
- Any such source MAY redirect device to a Bootstrap server.
  - Bootstrap server protocol is JSON or XML over HTTPS (RESTCONF)
- Secure (Zero Trust)
  - Mutually authenticated certificates: IDevID + Manufacturer's Trust Anchor
  - RFC 8366 Vouchers MAY be used to proxy trust from Manufacturer Authorized Signing Authority (MASA)
  - Bootstrapping data MAY be encrypted with Device's public key.

From Device's Perspective



Three Bootstrapping Artifacts

Only this artifact needed if transport-level security can be assured.
All three artifacts needed otherwise.

Conveyed Information

Redirect Information

- Tells bootstrapping device to look somewhere else.
- MAY convey a TLS certificate enabling device to establish Trust with a second location.

Onboarding Information

- Provides boot image details, initial configuration, and/or arbitrary scripts.

Ownership Voucher (from RFC 8366)

- Assigns device ownership to a “domain certificate”
- Public key used to authenticate the Owner Certificate

Owner Certificate

- Issued by the “domain certificate”
- Public key used to authenticate “signed data”

A 4th bootstrapping artifact?

In draft-ietf-netconf-sztp-csr, it becomes possible for the device to also obtain an LDevID certificate.

The LDevID can use the same public key as the IDevID or a fresh one with algorithms selected by the server.

Conveying Trust

- A device, in its factory default condition, can only trust certificates authorized by its Manufacturer (using trust anchors).

- Trusted anchor certificates are used in two ways:

    1. To authenticate that a remote TLS server' certificate is signed, somewhere in its chain, by the Manufacturer (or delegate).
    2. To verify that an RFC 8366 Voucher is signed by the Manufacturer (or delegate).

- If a source is NOT trusted, then response MUST be either:
    - An unsigned redirect response.
    - A signed response (i.e., using the Ownership Voucher + Owner Certificate).

SZTP + CSR (draft-ietf-netconf-sztp-csr - in Last Call)

Prestate:
- SZTP enabled
- IDevID cert

Bootstrapping Device

<Local Infrastructure>

Bootstrapping Server

CA

Power on

Discover a bootstrap server.
(removable storage, DNS, DHCP, Well-known service, etc.)

HTTPS: POST get-boostrapping-data

IDevID cert authenticated

- Connect to preconfigured address + port.
- Auth SZTPD's cert to preconfigured CA.
- Auth Self to SZTPD using IDevID.
- Send "csr-support" struct (identifying what algs are supported)

Discover that device needs an LDevID cert.

400 (Bad Request, containing selected algs)

HTTPS: POST get-boostrapping-data

- Connect again, sending "csr" structure (containing the CSR)

Relay CSR (or CMC, etc.) to CA

Sign CSR

LDevID

Stitch LDevID into config returned

200 (OK, Response body includes LDevID)

LDevID returned, along with rest of the onboarding response.

Comments or Questions?