NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 17, 2016

K. Watsen
Juniper Networks
M. Abrahamsson
T-Systems

March 16, 2016

Zero Touch Provisioning for NETCONF or RESTCONF based Management
draft-ietf-netconf-zerotouch-07

Abstract

This draft presents a secure technique for establishing a NETCONF or RESTCONF connection between a newly deployed device, configured with just its factory default settings, and its deployment specific network management system (NMS).

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. Please note that no other RFC Editor instructions are specified anywhere else in this document.

This document contains references to other drafts in progress, both in the Normative References section, as well as in body text throughout. Please update the following references to reflect their final RFC assignments:

- draft-ietf-netconf-call-home
- draft-ietf-netconf-restconf
- draft-ieft-netconf-server-model
- draft-pritikin-anima-bootstrapping-keyinfra

Artwork in this document contains shorthand references to drafts in progress. Please apply the following replacements:

- "XXXX" --> the assigned RFC value for this draft

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

- "2016-03-16" --> the publication date of this draft

The following one Appendix section is to be removed prior to publication:

o Appendix A. Change Log

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 17, 2016.

Copyright Notice

Table of Contents

1. Introduction

A fundamental business requirement for any network operator is to reduce costs where possible. For network operators, deploying devices to many locations can be a significant cost, as sending trained specialists to each site to do installations is both cost prohibitive and does not scale.

This document defines bootstrapping strategies enabling devices to securely obtain bootstrapping data with no installer input, beyond physical placement and connecting network and power cables. The ultimate goal of this document is to enable a secure NETCONF [RFC6241] or RESTCONF [draft-ietf-netconf-restconf] connection to the deployment specific network management system (NMS).

1.1. Use Cases

o Connecting to a remotely administered network

  This use-case involves scenarios, such as a remote branch office or convenience store, whereby a device connects as an access gateway to an ISP's network. Assuming it is not possible to customize the ISP's network to provide any bootstrapping support, and with no other nearby device to leverage, the device has no recourse but to reach out to an Internet-based bootstrap server to bootstrap off of.

o Connecting to a locally administered network

  This use-case covers all other scenarios and differs only in that the device may additionally leverage nearby devices, which may direct it to use a local service to bootstrap off of. If no such information is available, or the device is unable to

use the information provided, it can then reach out to network just as it would for the remotely administered network use-case.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in the sections below are to be interpreted as described in RFC 2119 [RFC2119].

This document uses the following terms:

Artifact: The term "artifact" is used throughout to represent the encoded form of any of Bootstrap Information, Redirect Information, Owner Certificate, and Ownership Voucher. The Bootstrap Server defined in this document is purposed to provide these artifacts, but they can also be provided by any other mechanism (removable storage, DHCP server, etc.), secure or not, so long as the principles for when the bootstrapping data needs to be signed is enforced.

Bootstrapping Data: The term "bootstrapping data" is used throughout this document to refer to the collection of data that a device may obtain from any source of bootstrapping data, including a removable storage device, a DHCP server, a DNS server, a Redirect Server, and/or a Bootstrap Server. This data includes both Redirect Information as well as Bootstrap Information.

Bootstrap Information: The term "bootstrap information" is used herein to refer to bootstrapping data that is used to guide a device to install a specific boot-image and commit a specific configuration. This data is formally defined by the "bootstrap-information" container in the YANG module defined in Section 7.4.

Bootstrap Server: The term "bootstrap server" is used within this document to mean any RESTCONF server implementing the YANG module defined in Section 7.4.

Device: The term "device" is used throughout this document to refer to the network element that needs to be bootstrapped. The device is the RESTCONF client to a Bootstrap Server (see above) and, at the end of bootstrapping process, the device is the NETCONF or RESTCONF server to a deployment-specific NMS. See Section 6 for more information about devices.

Initial Secure Device Identifier (IDevID): The term "IDevID" is defined in [Std-802.1AR-2009] as "the Secure Device Identifier

(DevID) installed on the device by the manufacturer". By example, an IDevID certificate, signed by the manufacturer may encode a manufacturer assigned unique identifier (e.g., serial number) and a public key matching a private key held within a TPM chip embedded within the device.

Network Management System (NMS): The acronym "NMS" is used throughout this document to refer to the deployment specific management system that the bootstrapping process is responsible for introducing devices to. From a device's perspective, when the bootstrapping process has completed, the NMS is a NETCONF or RESTCONF client.

Owner: See Rightful Owner.

Owner Certificate: The term "owner certificate" is used in this document to represent an X.509 certificate, signed by the device's manufacturer or delegate, that binds an owner identity to the owner's private key, which the owner can subsequently use to sign artifacts. The owner certificate is used by devices only when validating owner signatures on signed data. This data is formally defined by the "owner-certificate" container in the YANG module defined in Section 7.4.

Ownership Voucher: The term "ownership voucher" is used in this document to represent manufacturer-specific artifact, signed by the device's manufacturer or delegate, binding an owner identity (same as in the Owner Certificate) to one or more device identities (e.g., serial numbers). The ownership voucher is used by devices only when validating owner signatures on signed data. This data is formally defined by the "ownership-voucher" container in the YANG module defined in Section 7.4.

Redirect Information: The term "redirect information" is used herein to refer to bootstrapping data that redirects a device to connect to another Bootstrap Server. This data is formally defined by the "redirect-information" container in the YANG module defined in Section 7.4.

Redirect Server: The term "redirect server" is used to refer to a Bootstrap Server that only returns Redirect Information. A Redirect Server is particularly useful when hosted by a manufacturer, to redirect devices to a deployment-specific bootstrap server.

Rightful Owner: The term "rightful owner" is used herein to refer to the person or organization that purchased a device. Ownership is conveyed by a chain of trust established by a sequence of

authenticated secure connections and/or Signed Data, as described in Section 2.3.

Signed Data: The term "signed data" is used throughout to mean either Redirect Information or Bootstrap Information that has been signed by a device's Rightful Owner's private key. These artifacts MUST be signed whenever communicated using an unsecured mechanism. Any time data is signed, it MUST be presented along with an Owner Certificate and Ownership Voucher, which themselves do not need to be signed by the Rightful Owner's private key, as they already are signed by the manufacturer.

Unsigned Data: The term "unsigned data" is used throughout to mean either Redirect Information or Bootstrap Information that has not been signed by a device's Rightful Owner's private key. The option to use unsigned data MUST only be available only when the data is obtained over an authenticated secure connection, such as to a Bootstrap Server.

1.3. Tree Diagrams

A simplified graphical representation of the data models is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Braces "{" and "}" enclose feature names, and indicate that the named feature must be present for the subtree to be present.
- Abbreviations before data node names: "rw" (read-write) represents configuration data and "ro" (read-only) represents state data.
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Guiding Principles

This section provides overarching principles guiding the solution presented in this document.

2.1. Trust Anchors

A trust anchor is used in cryptography to represent an entity in which trust is implicit and not derived. In public key infrastructure using X.509 certificates, a root certificate is the trust anchor from which the chain of trust is derived. The solution presented in this document requires that all the entities involved possess specific trust anchors in order to ensure mutual authentication throughout the zero touch bootstrapping process.

2.2. Conveying Trust

A device in its factory default state possesses a limited set of manufacturer specified trust anchors. In this document, there are two types of trust anchors of interest. The first type of trust anchor is used to authenticate a secure connection to, for instance, a manufacturer-hosted Internet-based bootstrap server. The second type of trust anchor is used to authenticate manufacturer-signed data, such as the owner certificate and ownership voucher described in this document.

In the first case, trust is conveyed by the device first authenticating the secure connection to the server and then by the device trusting that the server would only provide data that its rightful owner staged for it to find. For instance, the staged data may be redirect information that includes the IP address and another trust anchor certificate for the deployment-specific bootstrap server. The device can then use the discovered trust anchor to authenticate a secure connection to the deployment-specific bootstrap server.

In the second case, trust is conveyed by the device first authenticating the owner certificate and ownership voucher and then, using the public key in the owner certificate, authenticate a signed artifact, such as redirect information. And again the device can use the discovered trust anchor to authenticate a secure connection to the deployment-specific bootstrap server.

2.3. Ownership

The goal of this document is to enable a device to connect with its rightful owner's NMS. This entails the manufacturer being able to track who owns which devices (out of the scope of this document), as well as an ability to convey that information to devices (in scope). Matching the two ways to convey trust, this document provides both a protocol oriented solution as well as an artifact based solution for conveying ownership.

The protocol based solution conveys ownership by the device first authenticating a secure connection to a bootstrap server and then trusting that the server would only provide data that its rightful owner staged for it to find. In the case of a manufacturer-hosted bootstrap server, the manufacturer takes the onus of ensuring that only data configured by the device's rightful owner is made available to the device. With this approach, the assignment of a device to an owner is ephemeral, with the manufacturer being able is reassign the device at any time.

The artifact based solution, which is ideal for when a secure connection cannot be established (e.g., loading data off a removable storage device), involves the manufacturer signing an owner certificate and then later, when the ownership for devices is established, the manufacturer signing a voucher that assigns those devices to the owner, and then the owner using their private key to sign the artifacts. Thus, from the device's perspective, it can use the presented ownership voucher to validate the presented owner certificate, which it can then use to validate the signature over the presented artifact. With this approach, the assignment of a device to an owner is somewhat permanent, as the ability for the manufacturer to reliably distribute CRLs to revoke assignments not possible when the devices do not contain a real time clock (see Section 8 for information about this).

3. Information Types

This document presumes there exists two types of zero touch information: redirect information and bootstrap information.

Both information types MAY be signed or unsigned, though in some contexts, as described below, the bootstrap information type MUST be signed, as there is not otherwise possible for a device to process it, even in a degraded manner.

Both information types MAY be encoded using various technologies. This document only tries to support the encodings supported by RESTCONF, namely XML and JSON, while leaving extensibility mechanisms in place to support future extensions.

3.1. Redirect Information

Redirect information provides a list of bootstrap servers, where each list entry includes the bootstrap server's hostname or IP address, an optional port, and an optional trust anchor certificate. The redirect information type is formally defined by the "redirect-information" grouping defined in Section 7.4.

As its name suggests, redirect information guides the device to attempt to connect to the specified bootstrap servers, until finding one that it can bootstrap itself off of. Redirect information is primarily distinguished from standard HTTP redirect by its optional inclusion of trust anchors, in which case it may be referred to as a "secure redirect".

Redirect information may be signed or unsigned. If the redirect information is not signed, then the device MUST NOT trust any included trust anchor certificates, equivalent to had they not been specified at all.

When redirect information is signed, then the device MUST establish a secure connection to the specified bootstrap server using X.509 certificate path validation ([RFC6125], Section 6) to the specified trust anchor, and MUST send its IDevID certificate in the form of a client certificate, and MUST POST notifications to the bootstrap server. Furthermore, in this case, any data obtained from the bootstrap server MAY NOT be signed, as it is already trusted by virtue of the secure connection.

When redirect information is unsigned, or doesn't specify a trust anchor certificate, and the device connects to the bootstrap server by blindly accepting the bootstrap server's TLS certificate, the device MUST NOT send its IDevID certificate in the form of a client certificate, and MUST NOT POST notifications to the bootstrap server. Furthermore, the device MUST assert that any data obtained from the bootstrap server is signed, much as it would assert bootstrap information loaded from a removable storage device is signed.

3.2. Bootstrap Information

Bootstrap information provides all the data neccessary for the device to bootstrap itself, in order to be considered ready to be managed. This data includes criteria about the boot image the device MUST be running, an initial configuration the device MUST commit, and an optional script that, if specified, the device MUST successfully execute. Descriptions for these follow:

- The boot image creteria is used to ensure the device is running a version of software that will be able to understand the configuration and script, if any. The criteria is flexible in that it allows for both an absolute specification of the boot image a device MUST be running, or just a list of YANG modules that the device MUST be able to understand.

- The configuration can configure any aspect of the device but, in order to fulfill the goal of the zero touch bootstrapping process,

to establish a NETCONF or RESTCONF connection to the device's deployment specific NMS, the configuration MUST minimally configure an administrator account (e.g., username, SSH public key) that the NMS can use to log into the device with, and configure the device to either listen for inbound NETCONF/RESTCONF connections, or for the device to initiate an outbound NETCONF/RESTCONF call home connection [draft-ietf-netconf-call-home]. The bootstrap information examples provided in Section 7.2.3, Section 7.2.4, and Section 7.3.2 all illustrate a minimal initial configuration.

- The script, if any, is used to perform non-configuration related activities deemed necessary. The script format is manufacturer specific. Requirements for scripts, such as exit status codes, are defined in the "script" node's description statement provided in the YANG module defined in Section 7.4.

Bootstrap information may be signed or unsigned. If the device is accessing the bootstrap server in an unsecured manner (e.g., from a removable storage device or from an untrusted server), then the bootstrap information MUST be signed, otherwise it MAY be signed.

Devices MUST process bootstrap information as is specified in Section 6.6.

The bootstrap information type is formally defined by the "bootstrap-information" grouping defined in Section 7.4.

4. Sources for Bootstrapping Data

Following are the sources of bootstrapping data that are referenced by the workflows presented in Section 5.3. Other sources of bootstrapping data may be defined in future documents, so long as the principles for when the bootstrapping data needs to be signed are enforced.

Each of the descriptions below show how the bootstrapping data needs to be handled in a manner consistent with the guiding principles in Section 2.

For devices supporting more than one source for bootstrapping data, no particular sequencing order has to be observed, as each source is equally secure, in that the chain of trust always goes back to the same root of trust, the manufacturer. That said, from a privacy perspective, it is RECOMMENDED that a device try to leverage local sources before remote source. For this reason, all the examples used in this document assume a removable storage device is accessed before

a DHCP server, which itself is accessed before an Internet-based bootstrap server.

4.1. Removable Storage

A device MAY attempt to acquire bootstrapping data from a directly attached removable storage device. The bootstrapping data MAY be either redirect information or bootstrap information.

If redirect information is provided, it SHOULD be signed, as removable storage devices are not trustworthy. However, if the redirect information is not signed, then the device MUST NOT trust any included trust anchor certificates, which means that the device would have to establish an unsecured connection to the specified bootstrap servers. See Section 3.1 for more about this case.

If bootstrap information is provided, it MUST be signed, as removable storage devices are not trustworthy and there is no option to process the data in a degraded manner, unlike as with redirect information.

For the case when the signed bootstrap information is provided, it is notable that even the raw boot image file itself can be on the removable storage device, by letting the URL reference a local file (e.g., file:///path/to/file), making use of the removable storage device a fully self-standing bootstrapping solution.

However, regardless if the boot image file resides on the local storage device or if the device must follow the URL to download it from a remote (and unsecured) server, the device MUST authenticate the validity of the boot image file, either by using the MD5 and SHA fingerprints supplied by the bootstrapping information, or by virtual of the boot image containing an embedded signature, if any.

4.2. DNS Server

A device MAY attempt to acquire bootstrapping data from a DNS server using DNS-based service discovery (DNS-SD) [RFC6763]. Due to DNS packet size limitations the bootstrapping data provided using DNS-SD can only be redirect information, no support for bootstrap information using DNS-SD is provided by this document.

The redirect information provided SHOULD be signed, as this document does not define a solution to secure the DNS records using DNSSEC [RFC6698]. However, if the redirect information is not signed, then the device MUST NOT trust any included trust anchor certificates, which means that the device would have to establish an unsecured connection to the specified bootstrap servers. See Section 3.1 for more about this case.

To use this approach, the device MAY perform DNS-SD via multicast DNS [RFC6762] searching for the service "_zerotouch._tcp.local.". Alternatively the device MAY perform DNS-SD via normal DNS operation, using the domain returned to it from the DHCP server, searching for the service "_zerotouch._tcp.example.com".

The mapping of redirect information onto DNS SRV [RFC2782] and DNS TXT [RFC1035] records as follows: is as follows:

- The bootstrap server's hostname or IP address is returned by the "Target" component of the DNS SRV record.

- The bootstrap server's port is returned by the "Port" component of the DNS SRV record.

- The bootstrap server's trust anchor is returned using the key "anchor" in the DNS TXT record with the binary value being the `gzip` compression over the redirect-information's "trust-anchor" value. To save additional space, it is RECOMMENDED that the trust anchor certificate uses an elliptical curve algorithm, rather than the seemingly ubiquitous RSA algorithm.

- The signature over the preceding three values is returned using the key "sig" in the DNS TXT record with the binary value being the `gzip` compression over the redirect-information's "signature" value.

- The owner certificate is returned using the key "cert" in the DNS TXT record with the binary value being the `gzip` compression over the redirect-information's "owner-certificate/certificate" value. There isn't enough space to support returning CRLs. To save additional space, it is RECOMMENDED that the owner certificate uses an elliptical curve algorithm, rather than the seemingly ubiquitous RSA algorithm.

- The ownership voucher is returned using the key "voucher" in the DNS TXT record binary value being the `gzip` compression over the redirect-information's "ownership-voucher/voucher" value. There isn't enough space to support returning CRLs.

The applicability of this approach across vendors is limited due to the ownership voucher being a manufacturer-specific format. This limitation only impacts signed data, when the ownership voucher is used; there is no such limitation when unsigned data is communicated.

4.3. DHCP Server

A device MAY attempt to acquire bootstrapping data from a DHCP server (e.g., using one of the DHCP options defined in Section 9.1). The bootstrapping data MAY be either redirect information or bootstrap information.

If redirect information is provided, it SHOULD be signed, as the DHCP protocol is not a secure protocol. However, if the redirect information is not signed, then the device MUST NOT trust any included trust anchor certificates, which means that the device would have to establish an unsecured connection to the specified bootstrap servers. See Section 3.1 for more about this case.

If bootstrap information is provided, it MUST be signed, as the DHCP protocol is not a secure protocol and there is no option to process the data in a degraded manner, unlike as with redirect information.

For the case when the signed bootstrap information is provided, it is notable that the URL would have to point to another file server (e.g., http://, ftp://, etc.), as DHCP servers do not themselves distribute files. In this case, the device MUST authenticate the validity of the boot image file, either by using the MD5 and SHA fingerprints supplied by the bootstrapping information, or by virtual of the boot image containing an embedded signature, if any.

It is expected that DHCP servers will provide redirect information more often than bootstrap information, since redirect information is more generic, potentially applicable to a large number of devices, with the number limited only by the number of devices listed by the associated ownership voucher. Still, because the ownership voucher is a manufacturer specific format, it is advisable for devices to send the Vendor Class Identifier (option 60) field in its DHCP lease request, so that the DHCP server doesn't accidentally hand it another manufacturer's voucher format.

If it is desired for the DHCP server to return bootstrap information, care should be taken to ensure that bootstrap information is applicable to all the devices that might connect to the DHCP server. The device SHOULD again pass the Vendor Class Identifier (option 60) field in its DHCP lease request. However, if it is desired to return device-specific bootstrap information, then the device SHOULD also send the Client Identifier (option 61) field in its DHCP lease request so that the DHCP server can select the specific bootstrap information that has been staged for that one device.

4.4. Bootstrap Server

A device MAY attempt to acquire bootstrapping data from a trusted Internet-based bootstrap server, a server implementing the RESTCONF API defined by the YANG module provided in Section 7.4. The bootstrapping data provided by the server MAY be either redirect information or bootstrap information.

Actually, a bootstrap server is not only a source for bootstrapping data, but it is also the consumer of notification messages from devices. These notification messages both enable visability into the bootstrapping process (e.g., reporting warnings and errors) and well as provide potentially useful completion status information (e.g., the device's SSH host-keys).

If the device is able to authenticate the bootstrap server, using X.509 certificate path validation ([RFC6125], Section 6) to a trust anchor the device was manufactured with, or it securely learned from another source of bootstrapping data, then the data the device obtains from the bootstrap server MAY NOT be signed. Notably, this is the only mechanism defined in this document whereby unsigned bootstrap information (not redirect information) can be used. When the device is able to authenticate the bootstrap server's TLS certificate, the device MUST send its IDevID certificate in the form of client-certificate and it MUST POST notifications to the bootstrap server.

If the device is unable to authenticate the bootstrap server's TLS certificate, for any reason, then any data it receives from the bootstrap server MUST be signed in order for the device to be able to make use of it. When the device is not able to authenticate the bootstrap server, the device MUST NOT send its IDevID in the form of a client-certificate and it MUST NOT POST any notifications to the bootstrap server.

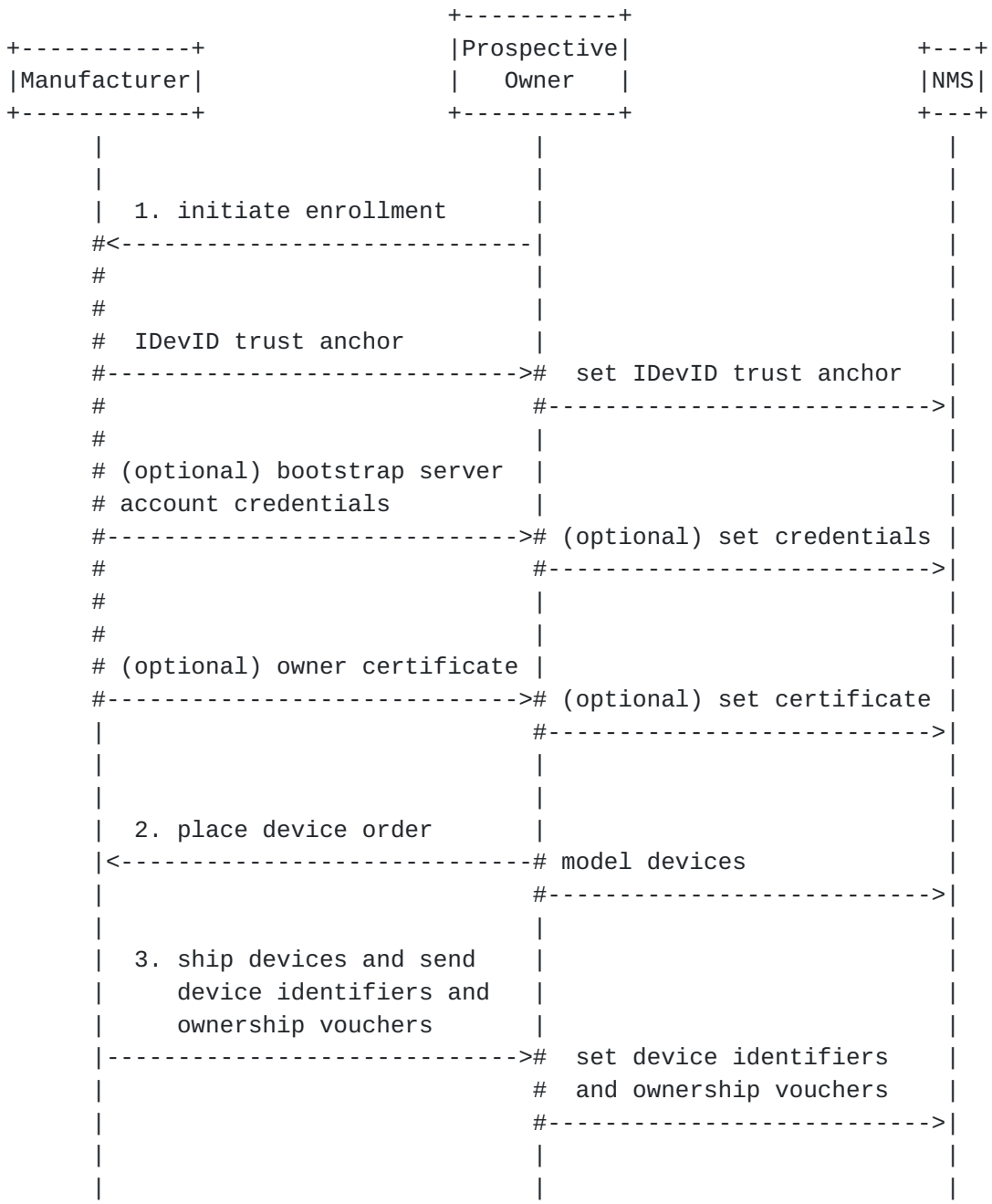
5. Workflow Overview

The zero touch solution presented in this document is conceptualized to be composed of the workflows described in this section. Implementations MAY vary in details. Each diagram is followed by a detailed description of the steps presented in the diagram, with further explanation on how implementations may vary.

5.1. Onboarding and Ordering Devices

The following diagram illustrates key interactions that occur from when a prospective owner enrolls in a manufacturer's zero touch

program to when the manufacturer ships devices for an order placed by the prospective owner.



The interactions in the above diagram are described below.

1. A prospective owner of a manufacturer's devices, or an existing owner that wishes to start using zero touch for future device orders, would initiate an enrollment process with the manufacturer, or the manufacturer's delegate.

2.

Regardless how the prospective owner intends to bootstrap their devices, they will always obtain from the manufacturer or delegate the trust anchor certificate needed to authenticate device IDevID certificates. This certificate will need to be installed on the prospective owner's NMS so that the NMS can subsequently authenticate the device's IDevID certificates.

If the manufacturer hosts an Internet based bootstrap server, such as described in Section 4.4, then credentials necessary to configure the bootstrap server would be provided to the prospective owner. If the bootstrap server is configurable through an API (outside the scope of this document), then the credentials might be installed on the prospective owner's NMS so that the NMS can subsequently configure the manufacturer-hosted bootstrap server directly.

If the manufacturer's devices are able to acquire bootstrapping data from sources other than a manufacturer-hosted Internet-based bootstrap server (e.g., removable storage, DHCP server, etc.), then the manufacturer would additionally provide an owner certificate to the prospective owner. How the owner certificate is used to enable devices to validate signed bootstrapping data is described in Section 6.4. Not depicted, the owner certificate is generated by the prospective owner previously sending a certificate signing request to the manufacturer for signing, thus resulting in the owner certificate. Assuming the prospective owner's NMS is able to prepare and sign the bootstrapping data, the owner certificate would be installed on the NMS at this time.

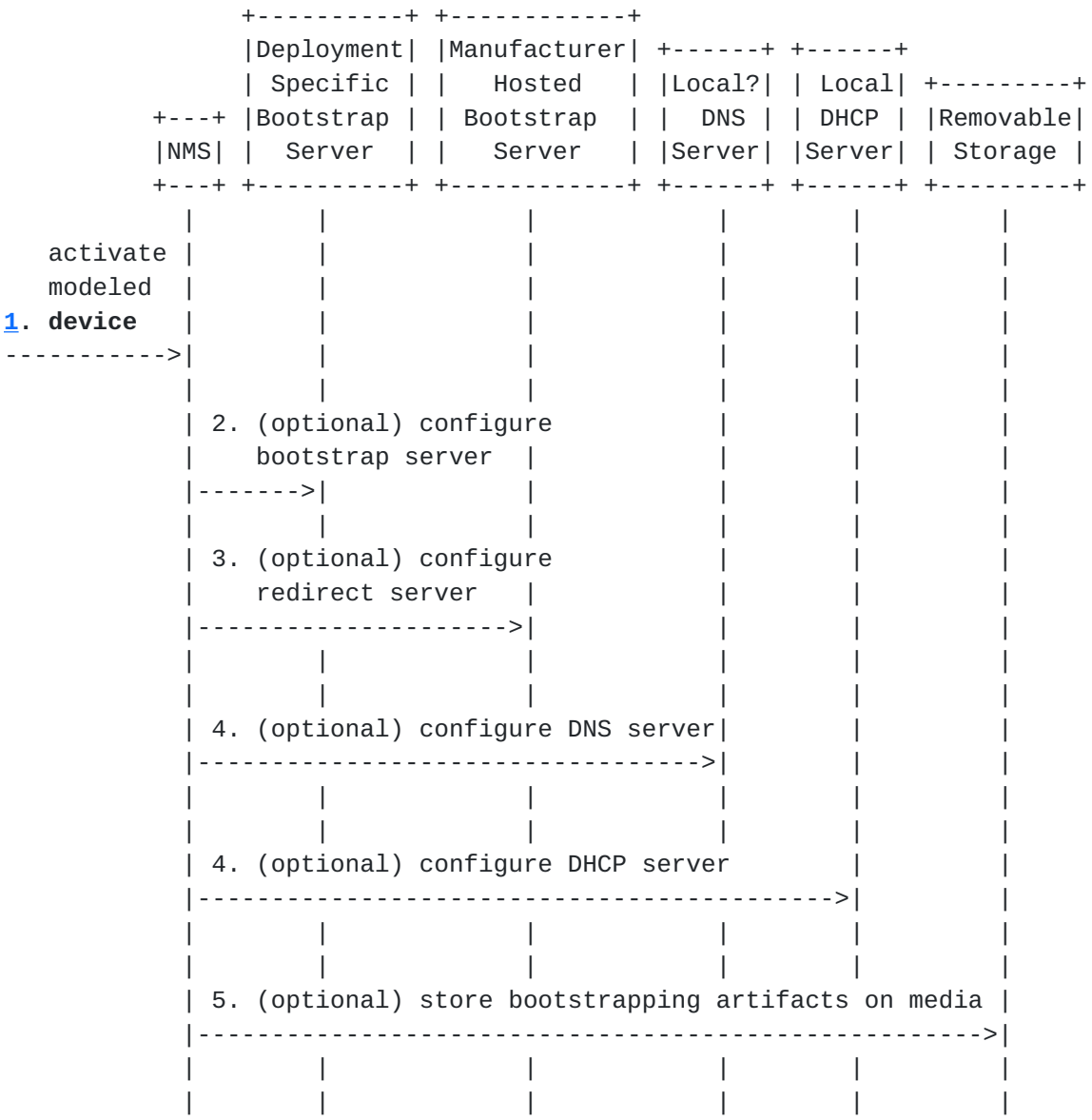
3. Some time later, the prospective owner places an order with the manufacturer, perhaps with a special flag checked for zero touch handling. At this time, or perhaps before placing the order, the owner may model the devices in their NMS. That is, create virtual objects for the devices with no real-world device associations. For instance the model can be used to simulate the device's location in the network and the configuration it should have when fully operational.

4. When the manufacturer ships the devices for the order, the manufacturer notifies the owner of the devices' unique identifiers and shipping destinations, which the owner can use to stage the network for when the devices powers on. Additionally, the manufacturer may send an ownership voucher, assigning

ownership of those devices to the rightful owner. The owner sets this information on their NMS, perhaps binding specific device identifiers and ownership vouchers (if supported) to specific modeled devices.

5.2. Owner Stages the Network for Bootstrap

The following diagram illustrates how an owner stages the network for bootstrapping devices.



The interactions in the above diagram are described below.

1. Having previously modeled the devices, including setting their fully operational configurations, associating device identifiers

and ownership vouchers (if supported), the owner "activates" one or more modeled devices. That is, tell the NMS to perform the steps necessary to prepare for when the real-world devices are powered up and initiate the bootstrapping process. Note that, in some deployments, this step might be combined with the last step from the previous workflow. Here it is depicted that an NMS performs the steps, but they may be performed manually or through some other mechanism.

2. If it is desired to use a deployment specific bootstrap server, it MUST be configured to provide the bootstrapping information for the specific devices. Whenever a deployment specific bootstrap server is used, the NMS MUST also configure some other source of bootstrapping data (i.e. an Internet based redirect server, a local DHCP server, a removable storage device, etc.) with redirect information, so that the device can discover where the deployment specific server is located and how to establish a connection to it. Configuring the bootstrap server MAY occur via a programmatic API not defined by this document. Illustrated here as an external component, the bootstrap server MAY be implemented as an internal component of the NMS itself.

3. If it is desired to use a manufacturer or delegate hosted bootstrap server, it MUST be configured to provide the bootstrapping information for the specific devices. The configuration MUST be either redirect or bootstrap information. That is, either the manufacturer hosted bootstrap server will redirect the device to another bootstrap server, or provide the device with its bootstrapping information itself. The types of bootstrapping information the manufacturer hosted bootstrap server supports MAY vary by implementation; some implementations may only support redirect information, or only support bootstrap information, or support both redirect and bootstrap information. Configuring the bootstrap server MAY occur via a programmatic API not defined by this document.

4. If it is desired to use a DNS server to supply bootstrapping information, a DNS server needs to be configured. If multicast DNS-SD is desired, then the server MUST reside on the local network, otherwise the MAY reside on a remote network. Please see Section 4.2 for more information about how to configure DNS servers. Configuring the DHCP server MAY occur via a programmatic API not defined by this document.

5. If it is desired to use a DHCP server to supply bootstrapping data, the DHCP server MUST be accessible via the network the device is located, either direct or via a DHCP relay. Please see Section 4.3 for more information about how to configure DHCP

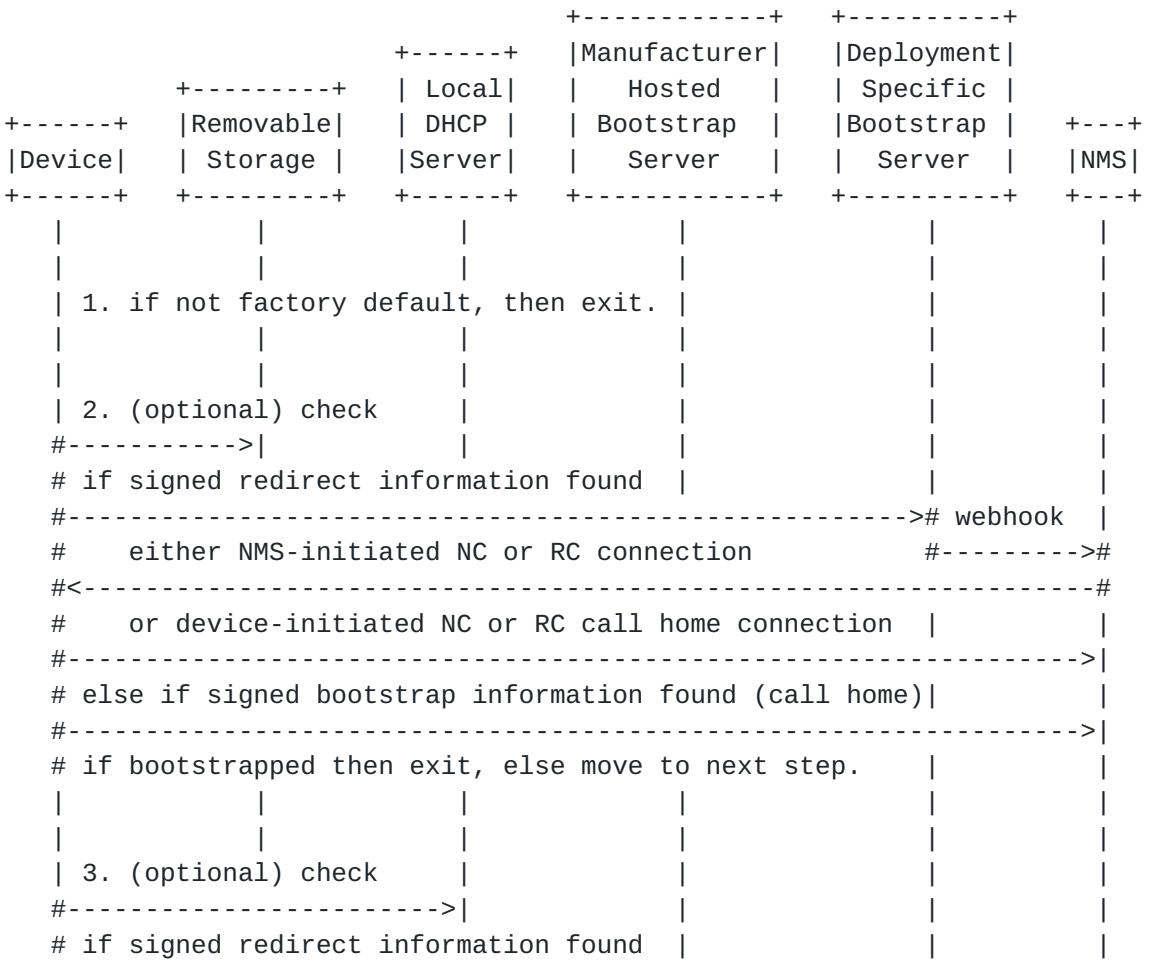
servers. Configuring the DHCP server MAY occur via a programmatic API not defined by this document.

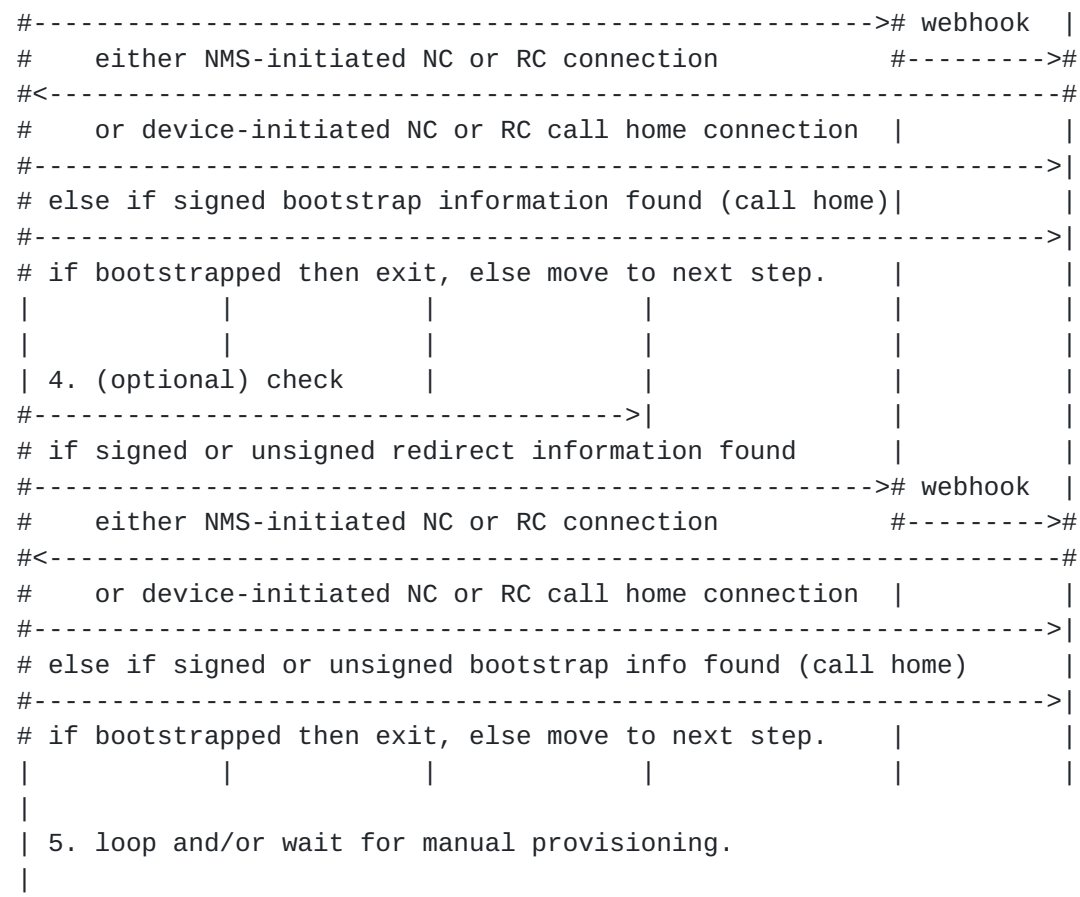
6. If it is desired to use a removable storage device (e.g., USB flash drive) to supply bootstrapping information, the information would need to be placed onto it. Please see Section 4.1 for more information about how to configure a removable storage device.

5.3. Device Powers On

The following diagram illustrates how a device might behave when powered on. Note that this is merely exemplary, subject to which bootstrapping strategies the device supports, which may be more or less than depicted below.

This diagram sequences the sources of bootstrapping information (see Section 4) based on locality, or how "close" the data is to the device, which is RECOMMENDED. Whether this sequence makes sense for a specific type of device needs to be determined by the manufacturer.





[Key: NC==NETCONF, RC==RESTCONF]

The interactions in the above diagram are described below.

1. Upon power being applied, the device's bootstrapping logic first checks to see if it is running in its factory default state. If it has a modified state, then the bootstrapping logic would exit and none to the following interactions would occur.

2. If the device is able to load bootstrapping data from a removable storage device (e.g., USB flash drive), it is RECOMMENDED that it try to do so first. Details such as the format of filesystem and the naming of the files are left to the device's manufacturer to define. Assuming a removable storage device is attached to the device, the device would check for bootstrapping data and, if found, validate that it has been signed using the procedure described in Section 6.4. The bootstrapping data MAY either be redirect information or bootstrap information. How the device processes each is follows:

   * In the case that redirect information is found (e.g., the example depicted in Section 7.3.1), the device would use the

redirect information to establish a secure connection to a deployment-specific bootstrap server. In theory this bootstrap server could return a response that redirected the device to yet another bootstrap server (e.g., the example depicted in Section 7.2.1), but in this example it is depicted that it returns bootstrap information (e.g., the example depicted in Section 7.2.3). Using this bootstrap information, the device would set its boot image and its initial configuration. If the bootstrap server supports notifying external systems (e.g., via a webhook) when a device has notified the bootstrap server that it is ready to be managed (e.g., the example depicted in Section 7.2.5), it might do so at this time, which could prompt the NMS to initiate a NETCONF or RESTCONF connection to the device at this time. Alternatively, the initial configuration the device installs could configure the device to initiate a NETCONF or RESTCONF call home [draft-ietf-netconf-call-home] connection to the deployment-specific NMS. All of these sub-steps are depicted in the diagram above.

* In the case that bootstrap information is found (e.g., the example depicted in Section 7.2.2), the device would use the bootstrap information to install a boot image, which itself could be located on the same removable storage device, and set its initial configuration. In this case, since there is no easy way to notify the NMS that the device is ready to be managed (e.g., via a webhook), it is RECOMMENDED that the initial configuration directs the device to proactively initiate a NETCONF or RESTCONF call home [draft-ietf-netconf-call-home] connection to the deployment-specific NMS.

If the device is unable to bootstrap using any of the information on the removable storage device, it would proceed to the next source of bootstrapping information, if any.

3. If the device is able to load bootstrapping data from a DHCP server, when obtaining a DHCP assignment, it may receive a response that includes a Zero Touch Information DHCP option (Section 9.1).

* If the redirect information contained in the DHCP option is signed, then it is RECOMMENDED that the device establish a secure TLS connection to the bootstrap server, by authenticating its TLS server certificate using the provided trust anchor, and download any data that has been staged for it there, which MAY not be signed, since the server's certificate could be trusted.

* On the other hand, if the redirect information contained in the DHCP option is unsigned, then it is RECOMMENDED that the device establish a unsecured TLS connection to the bootstrap server, by blindly accepting its TLS server certificate, and download any data that has been staged for it there, which then MUST be signed, since the server's certificate could not be trusted.

In either case, the remainder of the device's logic is the same as described above for when using a removable storage device. If the device is unable to bootstrap using information provided by a DHCP server, it would proceed to the next source of bootstrapping information, if any.

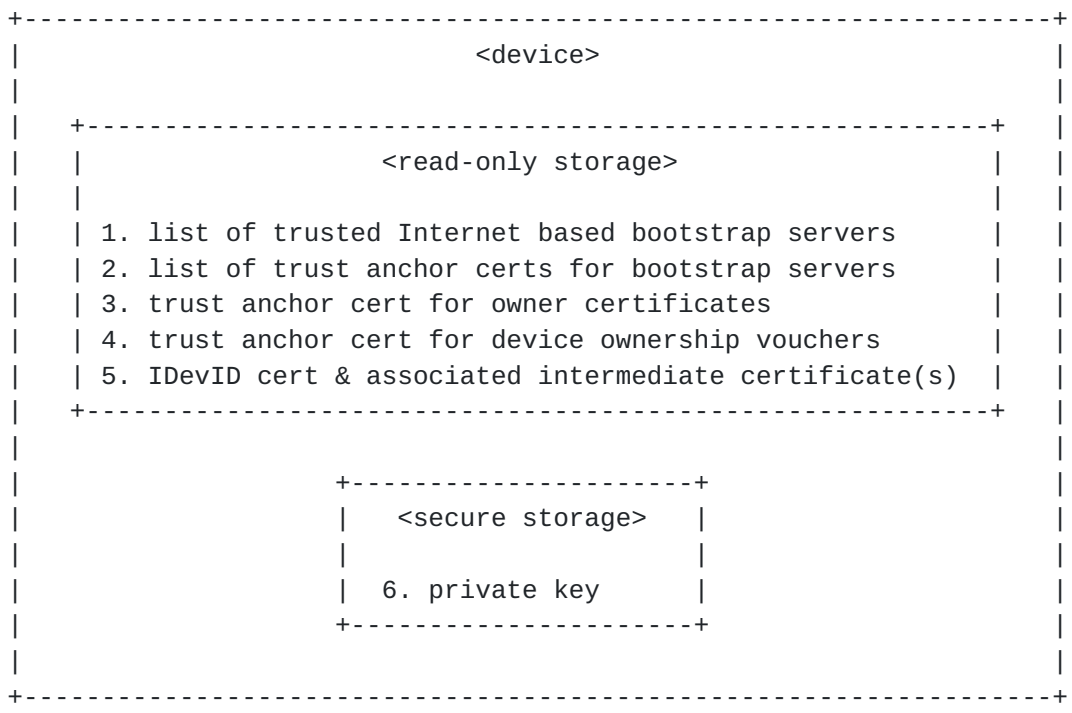
4. If the device is able to load bootstrapping data from a trusted Internet-based bootstrap server, as preconfigured in its factory default settings (Section 6.1), it is RECOMMENDED that the device attempts to establish a secure TLS connection to the bootstrap server, authenticating its TLS server certificate using the trust anchors set by its factory default state (Section 6.1), and download any data that has been staged for it there, which MAY not be signed, since the server's certificate could be trusted. In either case, the remainder of the device's logic is the same as described above for when using a removable storage device. If the device is unable to bootstrap using information provided by a DHCP server, it would proceed to the next source of bootstrapping information, if any.

5. If no more sources of bootstrapping information are available, the device MAY retry again all sources of bootstrapping data and/or MAY provide manageability interfaces for manual configuration (e.g., CLI, HTTP, NETCONF, etc.). If manual configuration is allowed, and such configuration is provided, the device MUST immediately cease trying to obtain bootstrapping data, as it would then no longer be in its factory default state.

6. Device Details

Devices supporting Zero Touch MUST have the preconfigured factory default state and bootstrapping logic described in the following sections.

6.1. Factory Default State



Each numbered item below corresponds to a numbered item in the diagram above.

1. Devices that support loading bootstrapping data from an Internet-based bootstrap server (see Section 4) MUST be manufactured with a list of trusted bootstrap servers. Each bootstrap server MAY be identified by just its hostname or IP address, and an optional port. Note that it is not necessary to configure URLs, as the RESTCONF protocol defines how the bootstrap server API specified in Section 7.4 maps into URLs.

2. Devices that support loading bootstrapping data from an Internet-based bootstrap server (see Section 4) SHOULD be manufactured with a list of trust anchor certificates that can be for X.509 certificate path validation [RFC6125], Section 6) on the bootstrap server's TLS server certificate.

3. Devices that support loading owner signed data (see Section 1.2) MUST be manufactured with the trust anchor certificate for the owner certificates that the manufacturer provides to prospective owners when they enroll in the manufacturer's Zero Touch program (see Section 5.1).

4. Devices that support loading owner signed data (see Section 1.2) MUST also be manufactured with the trust anchor certificate for the device ownership vouchers that the manufacturer provides to

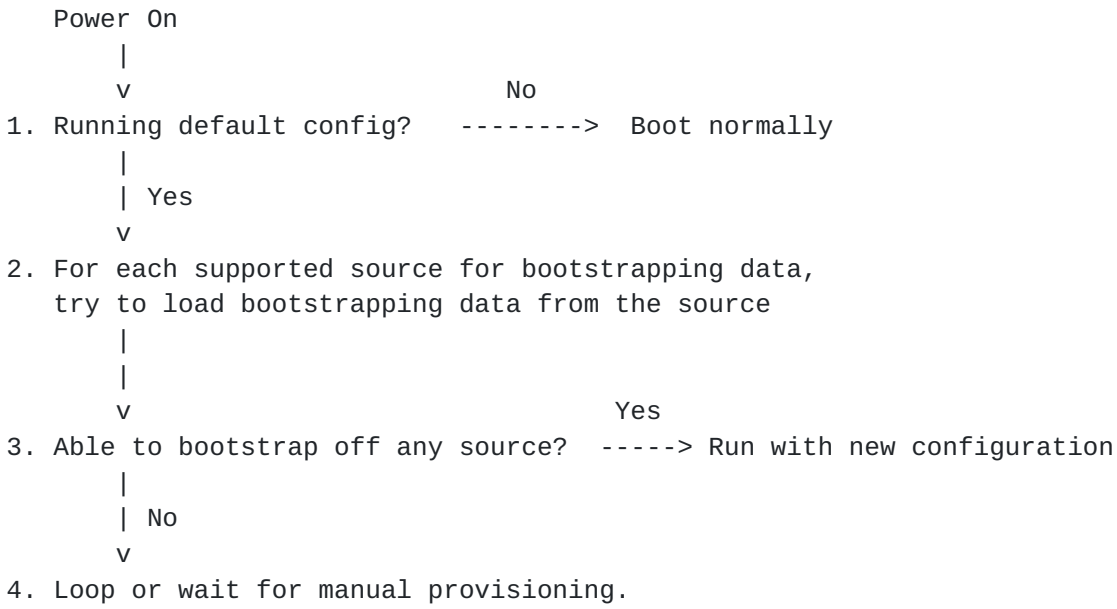
prospective owners when it ships out an order of Zero Touch devices (see Section 5.1).

5. Devices MUST be manufactured with an initial device identifier (IDevID), as defined in [Std-802.1AR-2009]. The IDevID is an X.509 certificate, encoding a unique device identifier (e.g., serial number). The device MUST also possess any intermediate certificates between the IDevID certificate and the manufacturer's IDevID trust anchor certificate.

6. Device MUST be manufactured with a private key that corresponds to the public key encoded in the device's IDevID certificate. This private key SHOULD be securely stored, ideally by a cryptographic processor (e.g., a TPM).

6.2. Boot Sequence

A device claiming to support Zero Touch MUST support the boot sequence described in this section.



These interactions are described next.

1. When the device powers on, it first checks to see if it is running the factory default configuration. If it is running a modified configuration, then it boots normally.

2. The device iterates over its list of sources for bootstrapping data Section 4. Details for how to processes a source of bootstrapping data are provided in Section 6.3.

3. If the device is able to bootstrap itself off any of the sources for bootstrapping data, it runs with the new bootstrapped configuration.

4. Otherwise the device MAY loop back through the list of bootstrapping sources again and/or wait for manual provisioning.

6.3. Processing a Source of Boostrapping Data

This section describes a recursive algorithm that a device claiming to support Zero Touch MUST use to authenticate bootstrapping data. A device enters this algorithm for each new source of bootstrapping data. The first time the device enters this algorithm, it MUST initialize a conceptual trust state variable, herein referred to as "trust-state", to FALSE. The ultimate goal of this algorithm is for the device to process bootstrap information (not redirect information) while its trust-state is TRUE.

If the data source is a bootstrap server, and the device is able to authenticate the server using X.509 certificate path validation ([RFC6125], Section 6) to one of the the device's preconfigured trust anchors, or to a trust anchor that it learned from a previous step, then the device MUST set trust-state to TRUE. If trust-state is TRUE, when connecting to the bootstrap server, the device MUST use its IDevID certificate for a client-certificate based authentication and MUST POST progress notifications using the bootstrap server's "notification" action. Otherwise, if trust-state is FALSE, when connecting to the bootstrap server, the device MUST NOT use its IDevID certificate for a client-certificate based authentication and MUST NOT POST progress notifications using the bootstrap server's "notification" action. When accessing a bootstrap server, the device MUST only access its top-level resource, to obtain all the data staged for it in one GET request, so that it can determine if the data is signed or not, and thus act accordingly.

For any data source, if the data is signed (i.e. the data includes a 'signature' field) and the device is able to validate the signed data using the algorithm described in Section 6.4, then the device MUST set trust-state to TRUE, else the device MUST set trust-state to FALSE. Note, this is worded to cover the special case when signed data is returned even from a trusted bootstrap server.

If the data is bootstrap information (not redirect information), and trust-state is FALSE, the device MUST exit the recursive algorithm, returning to the state machine described in Section 6.2. Otherwise, the device MUST attempt to process the bootstrap information as described in Section 6.6. In either case, success of failure, the device MUST exit the recursive algorithm, returning to the state

machine described in Section 6.2, the only difference being in how it responds to the "Able to bootstrap off any source?" conditional described in that state machine.

If the data is redirect information, the device MUST process the redirect information as described in Section 6.5. This is the recursion step, it will cause to device to reenter this algorithm, but this time the data source will most definitely be a bootstrap server, as that is all redirect information is able to do, though it's interesting to note that the bootstrap server's response MAY be more redirect information.

6.4. Validating Signed Data

Whenever a device is presented signed data, it MUST validate the signed data as described in this section.

Whenever there is signed data, the device MUST also be provided an ownership voucher and an owner certificate. How all the needed records are provided for each source of bootstrapping data is defined in Section 4

The device MUST first authenticate the ownership voucher by validating the signature on it to one of its preconfigured trust anchors (see Section 6.1) and verify that the voucher contains the device's unique identifier (e.g., serial number). If the authentication of the voucher is successful, the device extracts the Rightful owner's identity from the voucher for use in the next step.

Next the device MUST authenticate the owner certificate by performing X.509 certificate path validation on it to one of its preconfigured trust anchors (see Section 6.1) and by verifying that the Subject contained in the certificate matches the Rightful owner identity extracted from the voucher in the previous step. If the authentication of the certificate is successful, the device extracts the owner's public key from the certificate for use in the next step.

Finally the device MUST authenticate the signed data by verifying the signature on it was generated by the private key matching the public key extracted from the owner certificate in the previous step.

If any of these steps fail, then the device MUST mark the data as invalid and not perform any of the subsequent steps.

6.5. Processing Redirect Information

In order to process redirect information (Section 3.1), the device MUST follow the steps presented in this section.

Processing redirect information is straightforward. Essentially the device MUST immediately attempt to establish a RESTCONF connection to the provided bootstrap server IP address or hostname.

If a hostname is provided, and its DNS resolution is to more than one IP address, the device MUST attempt to try to connect to all of them, sequentially, until it is able to successfully bootstrap off one of them.

If the redirect information includes a trust anchor, and the redirect information can be trusted (e.g., trust-state is TRUE), then the device MUST authenticate the bootstrap server using X.509 certificate path validation ([RFC6125], Section 6) using the specified trust anchor.

6.6. Processing Bootstrap Information

In order to process bootstrap information (Section 3.2), the device MUST follow the steps presented in this section.

When processing bootstrap information, the device MUST first process the boot image information, then commit the initial configuration, and then execute the script, if any, in that order. If the device encounters an error at any step, it MUST NOT proceed to the next step.

First the device MUST determine if the image it is running satisfies the specified "boot-image" criteria. If it does not, the device MUST download and install the specified boot image, and reboot. Upon rebooting, the device MUST still be in its factory default state, causing the bootstrapping process to run again, which will eventually come to this very point, but this time the device's running image will satisfy the specified criteria, and thus the device moves to processing the next step.

Next the device commits the provided initial configuration. Assuming no errors, the device moves to processing the next step.

Next, for devices that support executing scripts, if a script has been specified, the device executes the script, checking its exit status code to determine if it succeeded, had warning, or had errors. In the case of errors, the device MUST reset itself in such a way

that force the reinstallation of its boot image, thereby wiping out any bad state the script might have left behind.

At this point, the device has completely processed the bootstrapping data and is ready to be managed. If the configuration configured the device it initiate a call home connection, it should proceed to do so now. Otherwise, the device should wait for a NETCONF or RESTCONF client to connect to it.

7. YANG-defined API and Artifacts

Central to the solution presented in this document is the use of a YANG module [RFC6020] to simultaneously define a RESTCONF based API for a bootstrap/redirect server as well as the encoding for signed artifacts that can be conveyed outside of the RESTCONF protocol (DHCP, FTP, TFTP, etc.).

The module defined in this section makes extensive use of data types defined in [RFC2315], [RFC5280], [RFC6991], and [RFC5280].

7.1. Module Overview

The following tree diagram Section 1.3 provides an overview for both the API and artifacts that can be used outside of RESTCONF.

```
module: ietf-zerotouch-bootstrap-server
   +--ro devices
      +--ro device* [unique-id]
         +--ro unique-id               string
         +--ro (type)?
         |  +--:(redirect-information)
         |  |  +--ro redirect-information
         |  |     +--ro bootstrap-server* [address]
         |  |        +--ro address         inet:host
         |  |        +--ro port?           inet:port-number
         |  |        +--ro trust-anchor    binary
         |  +--:(bootstrap-information)
         |     +--ro bootstrap-information
         |        +--ro boot-image
         |        |  +--ro modules
         |        |  |  +--ro module*
         |        |  |     +--ro name?       yang:yang-identifier
         |        |  |     +--ro revision?   string
         |        |  +--ro name       string
         |        |  +--ro md5        string
         |        |  +--ro sha1       string
         |        |  +--ro uri*       inet:uri
         |        +--ro configuration
         |        +--ro script?          string
         +--ro owner-certificate
         |  +--ro certificate    binary
         |  +--ro issuer-crl?    binary
         +--ro ownership-voucher
         |  +--ro voucher       binary
         |  +--ro issuer-vrl?   binary
         +--ro signature?               binary
         +---x notification
            +---w input
               +---w notification-type    enumeration
               +---w message?             string
               +---w ssh-host-keys
               |  +---w ssh-host-key*
               |     +---w format      enumeration
               |     +---w key-data    string
               +---w trust-anchors
                  +---w trust-anchor*
                     +---w protocol*      enumeration
                     +---w certificate    binary
```

In the above diagram, notice that all of the protocol accessible node are read-only, to assert that devices can only pull data from the bootstrap server.

Also notice that the module defines an action statement, which devices may use to provide progress notifications to the bootstrap server.

7.2. API Examples

This section presents some examples illustrating device interactions with a bootstrap server to access Redirect and Bootstrap information, both unsigned and signed, as well as to send a progress notification. These examples show the bootstrap information containing configuration defined by [RFC7317] and [draft-ietf-netconf-server-model].

7.2.1. Unsigned Redirect Information

The following example illustrates a device using the API to fetch its bootstrapping data. In this example, the device receives unsigned redirect information. This example is representative of a response a trusted redirect server might return.

```
REQUEST
-------
['\' line wrapping added for formatting only]

GET https://example.com/restconf/data/ietf-zerotouch-bootstrap-server:\
devices/device=123456 HTTP/1.1
HOST: example.com
Accept: application/yang.data+xml


RESPONSE
--------

HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+xml

<!-- '\' line wrapping added for formatting purposes only -->

<device
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <unique-id>123456789</unique-id>
  <redirect-information>
    <bootstrap-server>
      <address>phs1.example.com</address>
      <port>8443</port>
      <trust-anchor>
```

```
        WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
        lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
        zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
        NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
        VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
        V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
        NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
        Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
        WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
        QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
        MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
        25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
        RJSUJQFRStS0Cg==
      </trust-anchor>
    </bootstrap-server>
    <bootstrap-server>
      <address>phs2.example.com</address>
      <port>8443</port>
      <trust-anchor>
        WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
        lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
        zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
        NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
        VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
        V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
        NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
        Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
        WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
        QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
        MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
        25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
        RJSUJQFRStS0Cg==
      </trust-anchor>
    </bootstrap-server>
  </redirect-information>
</device>
```

7.2.2. Signed Redirect Information

The following example illustrates a device using the API to fetch its bootstrapping data. In this example, the device receives signed redirect information. This example is representative of a response that redirect server might return if concerned the device might not be able to authenticate its TLS certificate.

```
REQUEST
-------
['\' line wrapping added for formatting only]
```

```
GET https://example.com/restconf/data/ietf-zerotouch-bootstrap-server:\
devices/device=123456 HTTP/1.1
HOST: example.com
Accept: application/yang.data+xml


RESPONSE
--------

HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+xml

<!-- '\' line wrapping added for formatting purposes only -->

<device
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <unique-id>123456789</unique-id>
  <redirect-information>
    <bootstrap-server>
      <address>phs1.example.com</address>
      <port>8443</port>
      <trust-anchor>
        WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
        lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
        zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
        NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
        VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
        V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
        NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
        Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
        WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
        QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
        MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
        25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
        RJSUJQFRStS0Cg==
      </trust-anchor>
    </bootstrap-server>
    <bootstrap-server>
      <address>phs2.example.com</address>
      <port>8443</port>
      <trust-anchor>
        WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
        lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
        zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
        NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
        VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
```

```
        V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
        NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
        Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
        WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
        QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
        MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
        25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
        RJSUJQFRStS0Cg==
      </trust-anchor>
    </bootstrap-server>
  </redirect-information>
  <owner-certificate>
    <certificate>
      MIIExTCCA62gAwIBAgIBATANBgkqhkiG9w0BAQsFADCBqjELMAkGA1UEBhMCVVMx\
      EzARBgNVBAgTCkNhbGlmb3JuaWExEjAQBgNVBAcTCVN1bm55dmFsZTEZMBcGA1UE\
      ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
      Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
      QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
      ap1DgmS3IaYl/s4OOF8yzcYJprm8O7NyZp+Y9H1U/7Qfp97/KbqwCgkHSzOlnt0X\
      KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
      8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
      KmORbiKU2GTGZkaCgCjmrWpvrYWLoXv/sf2nPLyK6YjiWsslOJtRO+KzRbs2B18C\
      AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
      yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAOBgNVHQ8BAf8EBAMCAgQwQgYDVR0fBDswOTA3oDWgM4YxaHR0cDovL2NybC5q\
      dW5pcGVyLm5ldD9jYT1KdW5pcGVyX1RydXN0X0FuY2hvcl9DQTANBgkqhkiG9w0B\
      AQsFAAOCAQEAOuD7EBilqQcT3t2C4AXta1gGNNwdldLLw0jtk4BMiA9l//DZfskB\
      2AaJtiseLTXsMF6MQwDs1YKkiXKLu7gBZDlJ6NiDwy1UnXhi2BDG+MYXQrc6p76K\
      z3bsVwZlaJQCdF5sbggc1MyrsOu9QirnRZkIv3R8ndJH5K792ztLquulAcMfnK1Y\
      NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX7WJzEbT/G7MUfo\
      Sb+U2PVsQTDWEzUjVnG7vNWYxirnAOZ0OXEWWYxHUJntx6DsbXYuX7D1PkkNr7ir\
      96DpOPtX7h8pxxGSDPBXIyvg02aFMphstQ==
    </certificate>
    <issuer-crl>
      Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
```

```
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAO==
    </issuer-crl>
  </owner-certificate>
  <ownership-voucher>
    <voucher>
      ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
      Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAO
    </voucher>
    <issuer-crl>
      QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
      KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
      8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
      AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
    </issuer-crl>
  </ownership-voucher>
  <signature>
    RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
    QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
    MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
    NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
  </signature>
</device>
```

7.2.3. Unsigned Bootstrap Information

The following example illustrates a device using the API to fetch its bootstrapping data. In this example, the device receives unsigned bootstrapping information. This example is representative of a response a locally deployed bootstrap server might return.

```
REQUEST
-------
['\' line wrapping added for formatting only]
```

```
GET https://example.com/restconf/data/ietf-zerotouch-bootstrap-server:\
devices/device=123456 HTTP/1.1
HOST: example.com
Accept: application/yang.data+xml


RESPONSE
--------

HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+xml

<!-- '\' line wrapping added for formatting purposes only -->

<device
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <unique-id>123456789</unique-id>
  <bootstrap-information>
    <boot-image>
      <name>
        boot-image-v3.2R1.6.img
      </name>
      <md5>
        SomeMD5String
      </md5>
      <sha1>
        SomeSha1String
      </sha1>
      <uri>
        ftp://ftp.example.com/path/to/file
      </uri>
    </boot-image>
    <configuration>
      <!-- from ietf-system.yang -->
      <system xmlns="urn:ietf:params:xml:ns:yang:ietf-system">
        <authentication>
          <user>
            <name>admin</name>
            <ssh-key>
              <name>admin's rsa ssh host-key</name>
              <algorithm>ssh-rsa</algorithm>
              <key-data>AAAAB3NzaC1yc2EAAAADAQABAAABAQDeJMV8zrtsi8CgEsR\
              jCzfve2m6zD3awSBPrh7ICggLQvHVbPL89eHLuecStKL3HrEgXaI/O2Mw\
              E1lG9YxLzeS5p2ngzK61vikUSqfMukeBohFTrDZ8bUtrF+HMLlTRnoCVc\
              WAw1lOr9IDGDAuww6G45gLcHalHMmBtQxKnZdzU9kx/fL3ZS5G76Fy6sA\
              vg7SLqQFPjXXft2CAhin8xwYRZy6r/2N9PMJ2Dnepvq4H2DKqBIe340jW\
```

```
              EIuA7LvEJYql4unq4Iog+/+CiumTkmQIWRgIoj4FCzYkO9NvRE6fOSLLf\
              gakWVOZZgQ8929uWjCWlGlqn2mPibp2Go1</key-data>
            </ssh-key>
          </user>
        </authentication>
      </system>
      <!-- from ietf-netconf-server.yang -->
      <netconf-server
        xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server">
        <call-home>
          <application>
            <name>config-mgr</name>
            <ssh>
              <endpoints>
                <endpoint>
                  <name>east-data-center</name>
                  <address>11.22.33.44</address>
                </endpoint>
                <endpoint>
                  <name>west-data-center</name>
                  <address>55.66.77.88</address>
                </endpoint>
              </endpoints>
              <host-keys>
                <host-key>my-call-home-x509-key</host-key>
              </host-keys>
            </ssh>
          </application>
        </call-home>
      </netconf-server>
    </configuration>
  </bootstrap-information>
</device>
```

7.2.4. Signed Bootstrap Information

The following example illustrates a device using the API to fetch its bootstrapping data. In this example, the device receives signed bootstrap information. This example is representative of a response that bootstrap server might return if concerned the device might not be able to authenticate its TLS certificate.

```
REQUEST
-------

['\' line wrapping added for formatting only]

GET https://example.com/restconf/data/ietf-zerotouch-bootstrap-server:\
```

```
devices/device=123456 HTTP/1.1
HOST: example.com
Accept: application/yang.data+xml


RESPONSE
--------

HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+xml

<!-- '\' line wrapping added for formatting purposes only -->

<device
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <unique-id>123456789</unique-id>
  <bootstrap-information>
    <boot-image>
      <name>
        boot-image-v3.2R1.6.img
      </name>
      <md5>
        SomeMD5String
      </md5>
      <sha1>
        SomeSha1String
      </sha1>
      <uri>
        /path/to/on/same/bootserver
      </uri>
    </boot-image>
    <configuration>
      <!-- from ietf-system.yang -->
      <system xmlns="urn:ietf:params:xml:ns:yang:ietf-system">
        <authentication>
          <user>
            <name>admin</name>
            <ssh-key>
              <name>admin's rsa ssh host-key</name>
              <algorithm>ssh-rsa</algorithm>
              <key-data>AAAAB3NzaC1yc2EAAAADAQABAAABAQDeJMV8zrtsi8CgEsR\
              jCzfve2m6zD3awSBPrh7ICggLQvHVbPL89eHLuecStKL3HrEgXaI/O2Mw\
              E1lG9YxLzeS5p2ngzK61vikUSqfMukeBohFTrDZ8bUtrF+HMLlTRnoCVc\
              WAw1lOr9IDGDAuww6G45gLcHalHMmBtQxKnZdzU9kx/fL3ZS5G76Fy6sA\
              vg7SLqQFPjXXft2CAhin8xwYRZy6r/2N9PMJ2Dnepvq4H2DKqBIe340jW\
              EIuA7LvEJYql4unq4Iog+/+CiumTkmQIWRgIoj4FCzYkO9NvRE6fOSLLf\
```

```
            gakWVOZZgQ8929uWjCWlGlqn2mPibp2Go1</key-data>
          </ssh-key>
        </user>
      </authentication>
    </system>
    <!-- from ietf-netconf-server.yang -->
    <netconf-server
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server">
      <call-home>
        <application>
          <name>config-mgr</name>
          <ssh>
            <endpoints>
              <endpoint>
                <name>east-data-center</name>
                <address>11.22.33.44</address>
              </endpoint>
              <endpoint>
                <name>west-data-center</name>
                <address>55.66.77.88</address>
              </endpoint>
            </endpoints>
            <host-keys>
              <host-key>my-call-home-x509-key</host-key>
            </host-keys>
          </ssh>
        </application>
      </call-home>
    </netconf-server>
  </configuration>
</bootstrap-information>
<owner-certificate>
  <certificate>
    MIIExTCCA62gAwIBAgIBATANBgkqhkiG9w0BAQsFADCBqjELMAkGA1UEBhMCVVMx\
    EzARBgNVBAgTCkNhbGlmb3JuaWExEjAQBgNVBAcTCVN1bm55dmFsZTEZMBcGA1UE\
    ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
    Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
    QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
    MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
    ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
    RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
    ap1DgmS3IaYl/s4OOF8yzcYJprm8O7NyZp+Y9H1U/7Qfp97/KbqwCgkHSzOlnt0X\
    KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
    8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
    KmORbiKU2GTGZkaCgCjmrWpvrYWLoXv/sf2nPLyK6YjiWsslOJtRO+KzRbs2B18C\
    AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
    yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
    WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
```

```
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAOBgNVHQ8BAf8EBAMCAgQwQgYDVR0fBDswOTA3oDWgM4YxaHR0cDovL2NybC5q\
      dW5pcGVyLm5ldD9jYT1KdW5pcGVyX1RydXN0X0FuY2hvcl9DQTANBgkqhkiG9w0B\
      AQsFAAOCAQEAOuD7EBilqQcT3t2C4AXta1gGNNwdldLLw0jtk4BMiA9l//DZfskB\
      2AaJtiseLTXsMF6MQwDs1YKkiXKLu7gBZDlJ6NiDwy1UnXhi2BDG+MYXQrc6p76K\
      z3bsVwZlaJQCdF5sbggc1MyrsOu9QirnRZkIv3R8ndJH5K792ztLquulAcMfnK1Y\
      NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX7WJzEbT/G7MUfo\
      Sb+U2PVsQTDWEzUjVnG7vNWYxirnAOZ0OXEWWYxHUJntx6DsbXYuX7D1PkkNr7ir\
      96DpOPtX7h8pxxGSDPBXIyvg02aFMphstQ==
    </certificate>
    <issuer-crl>
      Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAO==
    </issuer-crl>
  </owner-certificate>
  <ownership-voucher>
    <voucher>
      ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
      Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
      GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
      X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
      MjAO
    </voucher>
    <issuer-vrl>
      QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
      MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
      ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
      RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
      KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
      8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
      AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
      WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
      NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
    </issuer-vrl>
```

```
  </ownership-voucher>
  <signature>
    RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
    QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
    MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
    NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
  </signature>
</device>
```

7.2.5. Progress Notifications

The following example illustrates a device using the API to post a notification to the server. The device may send more than one notification to the server (e.g., to provide status updates). The YANG module defines only one notification type, bootstrap-complete. Other notification types may be defined through YANG augmentation.

The bootstrap server MUST NOT process a notification from a device without first authenticating the device. This is in contrast to when a device is fetching data from the server, a read-only operation, in which case device authentication is not strictly required.

In this example, the device sends a notification indicating that it has completed bootstrapping off the data provided by the server. This example also illustrates the device sending its SSH host keys to the bootstrap server, which it might, for example, forward onto a downstream NMS component, so that the NMS can subsequently authenticate the device when establishing a NETCONF over SSH connection to it.

Note that the need for a device to provide its SSH host key (or TLS server certificate) in the "bootstrap-complete" message is unnecessary when the device is able to present its IDevID certificate [Std-802.1AR-2009] as its SSH host key or TLS server certificate, when subsequently establishing a NETCONF or RESTCONF connection with the deployment-specific NMS.

```
REQUEST
-------
['\' line wrapping added for formatting only]

POST https://example.com/restconf/data/ietf-zerotouch-bootstrap-server:\
devices/device=123456/notification HTTP/1.1
HOST: example.com
Content-Type: application/yang.data+xml

<!-- '\' line wrapping added for formatting purposes only -->
```

```
<input
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <notification-type>bootstrap-complete</notification-type>
  <message>example message</message>
  <ssh-host-keys>
    <ssh-host-key>
      <format>ssh-rsa</format>
      <key-data>
       AAAAB3NzaC1yc2EAAAADAQABAAABAQDeJMV8zrtsi8CgEsRCjCzfve2m6\
       zD3awSBPrh7ICggLQvHVbPL89eHLuecStKL3HrEgXaI/O2MwjE1lG9YxL\
       zeS5p2ngzK61vikUSqfMukeBohFTrDZ8bUtrF+HMLlTRnoCVcCWAw1lOr\
       9IDGDAuww6G45gLcHalHMmBtQxKnZdzU9kx/fL3ZS5G76Fy6sA5vg7SLq\
       QFPjXXft2CAhin8xwYRZy6r/2N9PMJ2Dnepvq4H2DKqBIe340jWqEIuA7\
       LvEJYql4unq4Iog+/+CiumTkmQIWRgIoj4FCzYkO9NvRE6fOSLLf6gakW\
       VOZZgQ8929uWjCWlGlqn2mPibp2Go1
      </key-data>
    </ssh-host-key>
    <ssh-host-key>
      <format>ssh-dsa</format>
      <key-data>
       zD3awSBPrh7ICggLQvHVbPL89eHLuecStKL3HrEgXaI/O2MwjE1lG9YxL\
       zeS5p2ngzK61vikUSqfMukeBohFTrDZ8bUtrF+HMLlTRnoCVcCWAw1lOr\
       9IDGDAuww6G45gLcHalHMmBtQxKnZdzU9kx/fL3ZS5G76Fy6sA5vg7SLq\
       AAAAB3NzaC1yc2EAAAADAQABAAABAQDeJMV8zrtsi8CgEsRCjCzfve2m6\
       QFPjXXft2CAhin8xwYRZy6r/2N9PMJ2Dnepvq4H2DKqBIe340jWqEIuA7\
       LvEJYql4unq4Iog+/+CiumTkmQIWRgIoj4FCzYkO9NvRE6fOSLLf6gakW\
       VOZZgQ8929uWjCWlGlqn2mPibp2Go1
      </key-data>
    </ssh-host-key>
  </ssh-host-keys>
  <trust-anchors>
    <trust-anchor>
      <protocol>netconf-ssh</protocol>
      <protocol>netconf-tls</protocol>
      <protocol>restconf-tls</protocol>
      <protocol>netconf-ch-ssh</protocol>
      <protocol>netconf-ch-tls</protocol>
      <protocol>restconf-ch-tls</protocol>
      <certificate>
        WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
        lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
        zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
        NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
        VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
        V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
        NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
        Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
        WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
```

```
        QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
        MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
        25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
        RJSUJQFRStS0Cg==
      </certificate>
    </trust-anchor>
  </trust-anchors>

</input>
```

```
RESPONSE
--------

HTTP/1.1 204 No Content
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
```

7.3. Artifact Examples

This section presents some examples for how the same information provided by the API can be packaged into stand alone artifacts. The encoding for these artifacts is the same as if an HTTP GET request had been sent to the RESTCONF URL for the specific resource. These examples show the bootstrap information containing configuration defined by [RFC7317] and [draft-ietf-netconf-server-model].

Encoding these artifacts for use outside of the RESTCONF protocol extends their utility for other deployment scenarios, such as when a local DHCP server or a removable storage device is used. By way of example, this may be done to address an inability for the device to access an Internet facing bootstrap/redirect server, or just for a preference to use locally deployed infrastructure.

7.3.1. Signed Redirect Information

The following example illustrates how a redirect can be encoded into an artifact for use outside of the RESTCONF protocol. The redirect information is signed so that it is secure even when no transport-level security is provided.

```
 <!-- '\' line wrapping added for formatting purposes only -->

 <redirect-information
    xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
   <bootstrap-server>
     <address>phs1.example.com</address>
     <port>8443</port>
     <trust-anchor>
       WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
       lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
       zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
       NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
       VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
       V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
       NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
       Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
       WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
       QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
       MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
       25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
       RJSUJQFRStS0Cg==
     </trust-anchor>
   </bootstrap-server>
   <bootstrap-server>
     <address>phs1.example.com</address>
     <port>8443</port>
     <trust-anchor>
       WmdsK2gyTTg3QmtGMjhWbW1CdFFVaWc3OEgrRkYyRTFwdSt4ZVRJbVFFM\
       lLQllsdWpOcjFTMnRLR05EMUc2OVJpK2FWNGw2NTdZNCtadVJMZgpRYjk\
       zSFNwSDdwVXBCYnA4dmtNanFtZjJma3RqZHBxeFppUUtTbndWZTF2Zwot\
       NGcEk3UE90cnNFVjRwTUNBd0VBQWFPQ0FSSXdnZ0VPCk1CMEdBMVVkRGd\
       VEJiZ0JTWEdlbUEKMnhpRHVOTVkvVHFLNWd4cFJBZ1ZOYUU0cERZd05ER\
       V6QVJCZ05WQkFNVENrTlNUQ0JKYzNOMVpYS0NDUUNVRHBNSll6UG8zREF\
       NQmdOVkhSTUJBZjhFCkFqQUFNQTRHQTFVZER3RUIvd1FFQXdJSGdEQnBC\
       Z05WSFI4RVlqQmdNRjZnSXFBZ2hoNW9kSFJ3T2k4dlpYaGgKYlhCc1pTN\
       WpiMjB2WlhoaGJYQnNaUzVqY215aU9LUTJNRFF4Q3pBSkJnTlZCQVlUQW\
       QmdOVkJBWVRBbFZUTVJBd0RnWURWUVFLRXdkbAplR0Z0Y0d4bE1RNHdEQ\
       MkF6a3hqUDlVQWtHR0dvS1U1eUc1SVR0Wm0vK3B0R2FieXVDMjBRd2kvZ\
       25PZnpZNEhONApXY0pTaUpZK2xtYWs3RTRORUZXZS9RdGp4NUlXZmdvN2\
   </bootstrap-server>
   <signature>
     RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
     QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
     MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
     NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
   </signature>
 </redirect-information>
```

7.3.2. Signed Bootstrap Information

The following example illustrates how bootstrapping data can be encoded into an artifact for use outside of the RESTCONF protocol. The bootstrap information is signed so that it is secure when no transport-level security is provided.

```
<!-- '\' line wrapping added for formatting purposes only -->

<bootstrap-information
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <boot-image>
    <name>
      boot-image-v3.2R1.6.img
    </name>
    <md5>
      SomeMD5String
    </md5>
    <sha1>
      SomeSha1String
    </sha1>
    <uri>
      file:///some/path/to/raw/file
    </uri>
  </boot-image>
  <configuration>
    <!-- from ietf-system.yang -->
    <system xmlns="urn:ietf:params:xml:ns:yang:ietf-system">
      <authentication>
        <user>
          <name>admin</name>
          <ssh-key>
            <name>admin's rsa ssh host-key</name>
            <algorithm>ssh-rsa</algorithm>
            <key-data>AAAAB3NzaC1yc2EAAAADAQABAAABAQDeJMV8zrtsi8CgEsRC\
            jCzfve2m6zD3awSBPrh7ICggLQvHVbPL89eHLuecStKL3HrEgXaI/O2Mwj\
            E1lG9YxLzeS5p2ngzK61vikUSqfMukeBohFTrDZ8bUtrF+HMLlTRnoCVcC\
            WAw1lOr9IDGDAuww6G45gLcHalHMmBtQxKnZdzU9kx/fL3ZS5G76Fy6sA5\
            vg7SLqQFPjXXft2CAhin8xwYRZy6r/2N9PMJ2Dnepvq4H2DKqBIe340jWq\
            EIuA7LvEJYql4unq4Iog+/+CiumTkmQIWRgIoj4FCzYkO9NvRE6fOSLLf6\
            gakWVOZZgQ8929uWjCWlGlqn2mPibp2Go1</key-data>
          </ssh-key>
        </user>
      </authentication>
    </system>
    <!-- from ietf-netconf-server.yang -->
    <netconf-server
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server">
```

```
      <call-home>
        <application>
          <name>config-mgr</name>
          <ssh>
            <endpoints>
              <endpoint>
                <name>east-data-center</name>
                <address>11.22.33.44</address>
              </endpoint>
              <endpoint>
                <name>west-data-center</name>
                <address>55.66.77.88</address>
              </endpoint>
            </endpoints>
            <host-keys>
              <host-key>my-call-home-x509-key</host-key>
            </host-keys>
          </ssh>
        </application>
      </call-home>
    </netconf-server>
  </configuration>
</bootstrap-information>
```

7.3.3. Owner Certificate

The following example illustrates how the owner certificate, along with its CRL, can be encoded into an artifact for use outside of the RESTCONF protocol. Note that the inclusion of the CLR is optional, and only present to support cases where the device is deployed on a private network, such that it would be unable to validate the revocation status of the certificate using an online lookup of the CRL or using OCSP. As the owner certificate and CRL are already signed by the manufacturer, an additional owner signature is unnecessary.

```
 <!-- '\' line wrapping added for formatting purposes only -->

 <owner-certificate
    xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
   <certificate>
     MIIExTCCA62gAwIBAgIBATANBgkqhkiG9w0BAQsFADCBqjELMAkGA1UEBhMCVVMx\
     EzARBgNVBAgTCkNhbGlmb3JuaWExEjAQBgNVBAcTCVN1bm55dmFsZTEZMBcGA1UE\
     ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
     Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
     QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
     MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
     ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
     RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
     ap1DgmS3IaYl/s4OOF8yzcYJprm8O7NyZp+Y9H1U/7Qfp97/KbqwCgkHSzOlnt0X\
     KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
     8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
     KmORbiKU2GTGZkaCgCjmrWpvrYWLoXv/sf2nPLyK6YjiWsslOJtRO+KzRbs2B18C\
     AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
     yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
     WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
     MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
     GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
     X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
     MjAOBgNVHQ8BAf8EBAMCAgQwQgYDVR0fBDswOTA3oDWgM4YxaHR0cDovL2NybC5q\
     dW5pcGVyLm5ldD9jYT1KdW5pcGVyX1RydXN0X0FuY2hvcl9DQTANBgkqhkiG9w0B\
     AQsFAAOCAQEAOuD7EBilqQcT3t2C4AXta1gGNNwdldLLw0jtk4BMiA9l//DZfskB\
     2AaJtiseLTXsMF6MQwDs1YKkiXKLu7gBZDlJ6NiDwy1UnXhi2BDG+MYXQrc6p76K\
     z3bsVwZlaJQCdF5sbggc1MyrsOu9QirnRZkIv3R8ndJH5K792ztLquulAcMfnK1Y\
     NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX7WJzEbT/G7MUfo\
     Sb+U2PVsQTDWEzUjVnG7vNWYxirnAOZ0OXEWWYxHUJntx6DsbXYuX7D1PkkNr7ir\
     96DpOPtX7h8pxxGSDPBXIyvg02aFMphstQ==
   </certificate>
   <issuer-crl>
     Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
     MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
     ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
     yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
     WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
     MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
     GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
     X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
     MjAO==
   </issuer-crl>
 </owner-certificate>
```

7.3.4. Ownership Voucher

The following example illustrates how the ownership voucher, along with its CRL, can be encoded into an artifact for use outside of the RESTCONF protocol. Note that the inclusion of the CLR is optional, and only present to support cases where the device is deployed on a private network, such that it would be unable to validate the revocation status of the certificate using an online lookup of the CRL or using OCSP. As the ownership voucher and CRL are already signed by the manufacturer, an additional owner signature is unnecessary.

```
<!-- '\' line wrapping added for formatting purposes only -->

<ownership-voucher
   xmlns="urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server">
  <voucher>
    ChQQSnVuaXBlcl9OZXR3b3JrczEdMBsGA1UECxQUQ2VydGlmaWNhdGVfSXNzdWFu\
    Y2UxGTAXBgNVBAMUEFRQTV9UcnVzdF9BbmNob3IxHTAbBgkqhkiG9w0BCQEWDmNh\
    MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
    ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
    yh/JaftWYf7m3KBzOdg2MIHfBgNVHSMEgdcwgdSAFDSljCNmTN5b+CDujJLlyDal\
    WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
    MBAGA1UEBxMJU3Vubnl2YWxlMRkwFwYDVQQKFBBKdW5pcGVyX05ldHdvcmtzMR0w\
    GwYDVQQLFBRDZXJ0aWZpY2F0ZV9Jc3N1YW5jZTEZMBcGA1UEAxQQVFBNX1RydXN0\
    X0FuY2hvcjEdMBsGCSqGSIb3DQEJARYOY2FAanVuaXBlci5jb22CCQDUbsEdTn5v\
    MjAO
  </voucher>
  <issuer-crl>
    QGp1bmlwZXIuY29tMB4XDTE0MDIyNzE0MTM1MloXDTE1MDIyNzE0MTM1MlowMDET\
    MBEGA1UEChQKVFBNX1ZlbmRvcjEZMBcGA1UEAxQQSnVuaXBlcl9YWFhYWF9DQTCC\
    ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5Mk5qFsVuqo+JmXWLmFxI\
    RDEuRiZNRNLeJpgN9YWkXLAZX2rASwy041EMmZ6KAkWUd3ZmXucfoLpdRemfuPii\
    KQTpIM/rNrbrkuTmalezFoFS7mrxLXJAsfP1guVcD7sLCyjvegL8pRCCrU9xyKLF\
    8u/Qz4s0x0uzcGYh0sd3iWj21+AtigSLdMD76/j/VzftQL8B1yp3vc1EZiowOwq4\
    AwEAAaOCAW0wggFpMBIGA1UdEwEB/wQIMAYBAf8CAQAwHQYDVR0OBBYEFHppoyXF\
    WFPaoYGwpIGtMIGqMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTES\
    NTOufhQsD2t4TYpEkzLEiZqSswdBOaPxPcJLQNW8Bw2xN+A9GX=
  </issuer-crl>
</ownership-voucher>
```

7.4. YANG Module

The bootstrap server's device-facing interface is normatively defined by the following YANG module:

```
<CODE BEGINS> file "ietf-zerotouch-bootstrap-server@2016-03-16.yang"
```

```
module ietf-zerotouch-bootstrap-server {
  yang-version "1.1";

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server";
  prefix "ztbs";

  import ietf-yang-types {     // RFC 6991
    prefix yang;
  }

  import ietf-inet-types {     // RFC 6991
    prefix inet;
  }

  organization
   "IETF NETCONF (Network Configuration) Working Group";

  contact
   "WG Web:   <http://tools.ietf.org/wg/netconf/>
    WG List:  <mailto:netconf@ietf.org>
    WG Chair: Mehmet Ersue
              <mailto:mehmet.ersue@nsn.com>
    WG Chair: Mahesh Jethanandani
              <mailto:mjethanandani@gmail.com>
    Editor:   Kent Watsen
              <mailto:kwatsen@juniper.net>";

  description
   "This module defines the southbound interface for Zero Touch
    bootstrap servers.

    Copyright (c) 2014 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD
    License set forth in Section 4.c of the IETF Trust's
    Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see
    the RFC itself for full legal notices.";

  revision "2016-03-16" {
    description
     "Initial version";
```

```
    reference
     "RFC XXXX: Zero Touch Provisioning for NETCONF Call Home";
  }


  container devices {
    config false;
    description
      "This is the top-level container for a device-facing protocol.
       As such it is read-only, how this data is configured is outside
       the scope of this data-model.  Further, it is expected that
       devices would only be able to access their data and not the
       data for any other device.";
    list device {
      key unique-id;

      description
        "A device's record entry.  This is the only RESTCONF resource
         that a device is expected to GET.  Getting this just this
         top-level provides the device with all the data it needs in
         a single request, which is ideal from both a performance and
         a resiliency perspectives..";

      leaf unique-id {
        type string;
        description
          "A unique identifier for the device (e.g., serial number).
           Each device accesses its bootstrapping record by its unique
           identifier.";
      }

      choice type {
        description
          "This choice statement ensures the response only contains
           redirect-information or bootstrap-information.";

        container redirect-information {
          description
            "This is redirect information data.  Its purpose is to
             redirect the device to another bootstrap server.  It
             contains a list of bootstrap servers.";

          list bootstrap-server {
            key address;
            description
              "A bootstrap server entry.";

            leaf address {
```

```
            type inet:host;
            description
             "The IP address or hostname of the bootstrap server
              the device should redirect to.";
          }
          leaf port {
            type inet:port-number;
            default 443;
            description
             "The port number the bootstrap server listens on.";
          }
          leaf trust-anchor {
            type binary;
            mandatory true;
            description
              "An X.509 v3 certificate structure as specified by RFC
               5280, Section 4 encoded using the ASN.1 distinguished
               encoding rules (DER), as specified in ITU-T X.690.  A
               certificate that a device can use as a trust anchor to
               authenticate the bootstrap server it is being redirected
               to.";
            reference
              "RFC 5280:
                 Internet X.509 Public Key Infrastructure Certificate
                 and Certificate Revocation List (CRL) Profile.
               ITU-T X.690:
                  Information technology - ASN.1 encoding rules:
                  Specification of Basic Encoding Rules (BER),
                  Canonical Encoding Rules (CER) and Distinguished
                  Encoding Rules (DER).";
          }
        }
      }

      container bootstrap-information {
        description
          "This is bootstrap information data.  Its purpose is to
           provide the device everything it needs to bootstrap
           itself.";

        container boot-image {
          description
            "Specifies criteria for the boot image the device MUST be
             running.";

          container modules {
            description
              "Specifies a list of YANG modules that the device MUST
```

```
          support.  This node is optional.  When this node is
          specified, the remaining nodes MUST be processed only
          in case the currently running image does not support
          any of the YANG modules, as a means to obtain a valid
          image.  When this node is not specified, then the
          device MUST ensure it is running the exact image, as
          specified by the remaining 'boot-image' nodes.";
        list module {
          description
            "Specifies a specific YANG modules, by its name and
             revision date.  The revision date is provided as a
             minimal revision date, and supported revision
             thereafter is considered sufficient";
          leaf name {
            type yang:yang-identifier;
            description
              "The YANG module's name.";
          }
          leaf revision {
            type string {
              pattern '\d{4}-\d{2}-\d{2}';
            }
            description
              "Represents a specific date in 2016-03-16 format.";
          }
        }
      }
      leaf name {
        type string;
        mandatory true;
        description
          "The name of a software image that either the device
           MUST be running, or MUST install only if its currently
           running image cannot support any of the required YANG
           modules.";
      }
      leaf md5 {
        type string;
        mandatory true;
        description
          "The hex-encoded MD5 hash over the boot-image file.";
      }
      leaf sha1 {
        type string;
        mandatory true;
        description
          "The hex-encoded SHA-1 hash over the boot-image file.";
      }
```

```
            leaf-list uri {
              type inet:uri;
              min-elements 1;
              description
                "An ordered list of URIs to where the boot-image file
                 may be obtained.  When the bootstrap information is
                 obtained from a bootstrap server, it is RECOMMENDED
                 that the list begins with absolute paths (e.g.,
                 beginning with '/') to the bootstrap server, so as
                 to leverage the existing secure connection.  If remote
                 URLs are also present in the list, deployments MUST
                 know in advance which URI schemes (https, http, ftp,
                 file, etc.) a device supports.  If a secure scheme
                 (e.g., https) is provided, devices MAY blindly accept
                 the server's credentials (e.g., TLS certificate).
                 Regardless how obtained, the device MUST ensure that
                 the boot-image is valid, either by leveraging a
                 signature embedded in the boot-image itself, if it
                 exists, or by first comparing the downloaded image to
                 both the MD5 and SHA1 fingerprints provided above.";
            }
          }

          anyxml configuration {   // pyang doesn't support anydata yet!
            description
              "Any configuration data model known to the device.  It may
               contain manufacturer-specific and/or standards-based data
               models.";
          }

          leaf script {
            type string;
            description
              "A device specific script that enables the execution of
               commands to perform actions not possible thru configuration
               alone.  The script SHOULD be executed with 'root' level
               permissions.

               If a script is erroneously provided to a device that
               does not support the execution of scripts, the device
               SHOULD send a 'script-warning' notification message,
               but otherwise continue processing the bootstrapping
               data as if the script had not been present.

               The script would return exit status code '0' on success
               and non-zero on error, with accompanying stderr/stdout
               for logging purposes.  In the case of an error, the exit
               status code will specify what the device should do.
```

```
            If the exit status code is greater than zero, then the
            device should assume that the script had soft failure
            that the script believes does not affect manageability.
            If the device obtained the bootstrap information from
            a bootstrap server, it SHOULD send a 'script-warning'
            notification message.

            If the exit status code is less than zero, the device
            should assume the script had a hard error that the
            script believes will affect manageability.  In this
            case, the device should try to send a 'script-error'
            notification message followed by a reset that will
            force a new boot-image install (wiping out anything
            the script may have done) and restart the entire
            bootstrapping process again.";
        }
      }
    }

    container owner-certificate {
      when "../ownership-voucher" {
        description
          "The owner certificate is only configurable when there
           also exists an ownership voucher.";
      }
      description
        "It is intended that the device will fetch this container
         as a whole, as it contains values that need to be
         processed together.";

      leaf certificate {
        type binary;
        mandatory true;
        description
          "An X.509 v3 certificate structure as specified by RFC
           5280, Section 4 encoded using the ASN.1 distinguished
           encoding rules (DER), as specified in ITU-T X.690.
           This certificate, signed by a manufacturer or delegate,
           for an owner, must encode a manufacturer-assigned value
           identifying the organization. This identifier must match
           the owner identifier encoded in the ownership voucher.";
        reference
          "RFC 5280:
             Internet X.509 Public Key Infrastructure Certificate
             and Certificate Revocation List (CRL) Profile.
           ITU-T X.690:
              Information technology - ASN.1 encoding rules:
              Specification of Basic Encoding Rules (BER),
```

```
            Canonical Encoding Rules (CER) and Distinguished
            Encoding Rules (DER).";
      }

      leaf issuer-crl {
        type binary;
        description
          "An CRL structure as specified by RFC 5280, Section 5
           encoded using the ASN.1 distinguished encoding rules
           (DER), as specified in ITU-T X.690.  The CRL for the
           CA that signed the owner certificate.  The CRL should
           be as up to date as possible.  This leaf is optional
           as it is only needed to support deployments where the
           device is unable to download the CRL from and of the
           distribution points listed in the owner certificate.";
        reference
          "RFC 5280:
             Internet X.509 Public Key Infrastructure Certificate
             and Certificate Revocation List (CRL) Profile.
           ITU-T X.690:
              Information technology - ASN.1 encoding rules:
              Specification of Basic Encoding Rules (BER),
              Canonical Encoding Rules (CER) and Distinguished
              Encoding Rules (DER).";
      }
    }

    container ownership-voucher {
      when "../signature" {
        description
          "An ownership voucher is only configurable when there
           also exists a signature.";
      }
      must "../owner-certificate" {
        description
          "An owner certificate must be present whenever an
           ownership voucher is present.";
      }
      description
        "This container contains the ownership voucher that the
         device uses to ascertain the identity of its rightful
         owner, as certified by its manufacturer.";

      leaf voucher {
        type binary;
        mandatory true;
        description
          "A manufacturer-specific encoding binding unique device
```

```
          identifiers to an owner identifier value matching the
          value encoded in the owner-certificate below.";
      }

      leaf issuer-vrl {
        type binary;
        description
          "An manufacturer-specific encoding of a voucher revocation
           list (VRL) for the issuer used by the manufacturer or
           delegate to sign ownership vouchers.  The VRL should be
           as up to date as possible.  This leaf is optional as it
           is only needed to support deployments where the device
           is unable to download the VRL from the manufacturer or
           delegate using some manufacturer-specific mechanism.";
      }
    }

    leaf signature {
      type binary;
      must "../ownership-voucher" {
        description
          "An ownership voucher must be present whenever an
           signature is present.";
      }
      description
        "A PKCS #7 SignedData structure as specified by RFC
         2315, Section 9.1 encoded using the ASN.1 distinguished
         encoding rules (DER), as specified in ITU-T X.690.
         This signature is generated using the owner's private
         private key and an owner-selected digest algorithm over
         the redirect-information or the bootstrap-information
         nodes, which ever is present, and in whatever encoding
         they are presented in (e.g., XML, JSON, etc.).";
         // is there a canonical format?
      reference
        "RFC 2315:
            PKCS #7: Cryptographic Message Syntax Version 1.5
         ITU-T X.690:
            Information technology - ASN.1 encoding rules:
            Specification of Basic Encoding Rules (BER),
            Canonical Encoding Rules (CER) and Distinguished
            Encoding Rules (DER).";
    }


    action notification {
      input {
        leaf notification-type {
```

```
            type enumeration {
              enum bootstrap-initiated {
                description
                  "Indicates that the device has just accessed
                   the bootstrap server.  The 'message' field
                   below SHOULD contain any additional information
                   that the manufacturer thinks might be useful,
                   or omitted entirely.";
              }
              enum validation-error {
                description
                  "Indicates that the device had an issue validating
                   the response from the bootstrap server.  The
                   'message' field below SHOULD indicate the specific
                   error.  This message also indicates that the device
                   has abandoned trying to bootstrap off this bootstrap
                   server.";
              }
              enum signature-validation-error {
                description
                  "Indicates that the device had an issue validating
                   the bootstrapping data.  For instance, this could
                   be due to the device expecting signed data, but
                   only found unsigned data, or because the ownership
                   voucher didn't include its unique identifier, or
                   because the signature didn't match, or and other
                   relevant error.  This 'message' field below SHOULD
                   indicate the specific error.  This message also
                   indicates that the device has abandoned trying to
                   bootstrap off this bootstrap server.";
              }
              enum image-mismatch {
                description
                  "Indicates that the device has determined that
                   its running image does not meet the specified
                   criteria.  The 'message' field below SHOULD
                   indicate both what image the device is currently
                   running as well as the criteria that failed.";
              }
              enum image-download-error {
                description
                  "Indicates that the device had an issue downloading
                   the image, which could be anything from the file
                   server being unreachable to the downloaded file
                   being the incorrect file (signature mismatch). The
                   'message' field about SHOULD indicate the specific
                   error.  This message also indicates that the device
                   has abandoned trying to bootstrap off this bootstrap
```

```
        server.";
      }
      enum config-warning {
        description
          "Indicates that the device obtained warning messages
           when it committed the initial configuration.  The
           'message' field below SHOULD indicate the warning
           messages that were generated.";
      }
      enum config-error {
        description
          "Indicates that the device obtained error messages
           when it committed the initial configuration.  The
           'message' field below SHOULD indicate the error
           messages that were generated.  This message also
           indicates that the device has abandoned trying to
           bootstrap off this bootstrap server.";
      }
      enum script-warning {
        description
          "Indicates that the device obtained a greater than
           zero exit status code from the script when it was
           executed.  The 'message' field below SHOULD indicate
           both the resulting exit status code and well as
           capture any stdout/stderr messages the script may
           have produced.";
      }
      enum script-error {
        description
          "Indicates that the device obtained a less than zero
           exit status code from the script when it was executed.
           The 'message' field below SHOULD indicate both the
           resulting exit status code and well as capture any
           stdout/stderr messages the script may have produced.
           This message also indicates that the device has
           abandoned trying to bootstrap off this bootstrap
           server.";
      }
      enum bootstrap-complete {
        description
          "Indicates that the device successfully processed the
           all the bootstrapping data and that it is ready to
           be managed.  The 'message' field below SHOULD contain
           any additional information that the manufacturer
           thinks might be useful, or omitted entirely.  At
           this point, the device is not expected to access
           the bootstrap server again.";
      }
```

```
            enum informational {
              description
                "Provided any additional information not captured by
                 any of the other notification-type.  The 'message'
                 field below SHOULD contain any additional information
                 that the manufacturer thinks might be useful, or
                 omitted entirely.";
            }
          }
          mandatory true;
          description
            "The type of notification provided.";
        }
        leaf message {
          type string;
          description
            "An optional human-readable value.";
        }
        container ssh-host-keys {
          description
            "A list of SSH host keys an NMS may use to authenticate
             a NETCONF connection to the device with.";
          list ssh-host-key {
            when "../type = bootstrap-complete" {
              description
                "SSH host keys are only sent when the notification
                 type is 'bootstrap-complete'.";
            }
            description
              "An SSH host-key";
            leaf format {
              type enumeration {
                enum ssh-dss { description "ssh-dss"; }
                enum ssh-rsa { description "ssh-rsa"; }
              }
              mandatory true;
              description
                "The format of the SSH host key.";
            }
            leaf key-data {
              type string;
              mandatory true;
              description
                "The key data for the SSH host key";
            }
          }
        }
        container trust-anchors {
```

```
            description
              "A list of trust anchor certificates an NMS may use to
               authenticate a NETCONF or RESTCONF connection to the
               device with.";
            list trust-anchor {
              when "../type = bootstrap-complete" {
                description
                  "Trust anchors are only sent when the notification
                   type is 'bootstrap-complete'.";
              }
              description
                "A list of trust anchor certificates an NMS may use to
                 authenticate a NETCONF or RESTCONF connection to the
                 device with.";
              leaf-list protocol {
                type enumeration {
                  enum netconf-ssh     { description "netconf-ssh"; }
                  enum netconf-tls     { description "netconf-tls"; }
                  enum restconf-tls    { description "restconf-tls"; }
                  enum netconf-ch-ssh  { description "netconf-ch-ssh"; }
                  enum netconf-ch-tls  { description "netconf-ch-tls"; }
                  enum restconf-ch-tls { description "restconf-ch-tls"; }
                }
                min-elements 1;
                description
                  "The protocols that this trust anchor secures.";
              }
              leaf certificate {
                type binary;
                mandatory true;
                description
                  "An X.509 v3 certificate structure as specified by RFC
                   5280, Section 4 encoded using the ASN.1 distinguished
                   encoding rules (DER), as specified in ITU-T X.690.";
                reference
                  "RFC 5280:
                     Internet X.509 Public Key Infrastructure Certificate
                     and Certificate Revocation List (CRL) Profile.
                   ITU-T X.690:
                      Information technology - ASN.1 encoding rules:
                      Specification of Basic Encoding Rules (BER),
                      Canonical Encoding Rules (CER) and Distinguished
                      Encoding Rules (DER).";
              }
            }
          }
        }
      } // end action
```

```
    }
  }
}
```

<CODE ENDS>

8. Security Considerations

8.1. Immutable storage for trust anchors

Devices MUST ensure that all their trust anchor certificates, including those for the owner certificate and ownership voucher, are protected from external modification.

It may be necessary to update these certificates over time (e.g., the manufacturer wants to delegate trust to a new CA). It is therefore expected that devices MAY update these trust anchors when needed through a verifiable process, such as a software upgrade using signed software images.

8.2. Clock Sensitivity

The solution in this document relies on TLS certificates, owner certificates, ownership vouchers, and CRLs, all of which require an accurate clock in order to be processed correctly. Devices implementations should take care to ensure the devices have a reliable clock when processing signed data, ideally be using a built-in real time clock (RTC). If a device does not have an RTC, then it SHOULD try to use NTP to initialize its clock before processing any time-sensitive bootstrapping data. It is understood that NTP is itself unsecured, not enabling the client to authenticate the server, and therefore easily spoofed. In the case that NTP is spoofed, it is possible for a replay attack to occur where an ownership voucher assignment from a previous owner is replayed on a device that has since been claimed by a new owner. For this reason, for devices that do not contain an RTC, it is RECOMMENDED that manufacturers only issue a single ownership voucher for the lifetime of a device.

8.3. Blindly authenticating a bootstrap server

This document allows a device to blindly authenticate a bootstrap server's TLS certificate. It does so to allow for cases where the redirect information may be obtained in an unsecured manner (e.g., via a DNS service discovery lookup, where only a hostname or IP address is returned).

To compensate for this, this document requires that devices do not send their IDevID certificate for client authentication, and that they do not POST any progress notifications, and that they assert that data downloaded from the server is signed, just as bootstrapping data would need to be signed if read from a removable storage device.

8.4. Entropy loss over time

Section 7.2.7.2 of the IEEE Std 802.1AR-2009 standard says that IDevID certificate should never expire (i.e. having a notAfter 99991231235959Z). Given the long-lived nature of these certificates, it is paramount to use a strong key length (e.g., 512-bit ECC).

8.5. Serial Numbers

This draft suggests using the device's serial number as the unique identifier in its IDevID certificate. This is because serial numbers are ubiquitous and prominently contained in invoices and on labels affixed to devices and their packaging. That said, serial numbers many times encode revealing information, such as the device's model number, manufacture date, and/or sequence number. Knowledge of this information may provide an adversary with details needed to launch an attack.

9. IANA Considerations

9.1. The BOOTP Manufacturer Extensions and DHCP Options Registry

The following registrations are in accordance to RFC 2939 [RFC2939] for "BOOTP Manufacturer Extensions and DHCP Options" registry maintained at http://www.iana.org/assignments/bootp-dhcp-parameters.

9.1.1. DHCP v4 Option

Tag: XXX

Name: Zero Touch Redirect Information

Returns a YANG-defined redirect-information object, encoded in the encoding specified by 'encoding'. Currently only "xml" and "json" are supported.

Code	Len	encoding	redirect-information
XXX	n		

Reference: RFC XXXX

9.1.2. DHCP v6 Option

Tag: YYY

Name: Zero Touch Redirect Information

Returns a YANG-defined redirect-information object, encoded in the encoding specified by 'encoding'. Currently only "xml" and "json" are supported.

Code	Len		
XXX	n	encoding	redirect-information

Reference: RFC XXXX

9.2. The IETF XML Registry

This document registers one URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested:

URI: urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server
Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

9.3. The YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC6020]. Following the format defined in [RFC6020], the the following registration is requested:

```
name:         ietf-zerotouch-bootstrap-server
namespace:    urn:ietf:params:xml:ns:yang:ietf-zerotouch-bootstrap-server
prefix:       ztbs
reference:    RFC XXXX
```

10. Other Considerations

Both this document and [draft-pritikin-anima-bootstrapping-keyinfra] define bootstrapping mechanisms. The authors have collaborated on both solutions and believe that each solution has merit and, in fact, can work together. That is, it is possible for a device to support both solutions simultaneously.

11. Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): David Harrington, Michael Behringer, Dean Bogdanovic, Martin Bjorklund, Joe Clarke, Toerless Eckert, Stephen Farrell, Stephen Hanna, Wes Hardaker, Russ Mundy, Reinaldo Penno, Randy Presuhn, Max Pritikin, Michael Richardson, Juergen Schoenwaelder.

Special thanks goes to Steve Hanna, Russ Mundy, and Wes Hardaker for brainstorming the original I-D's solution during the IETF 87 meeting in Berlin.

12. References

12.1. Normative References

[draft-ietf-netconf-call-home]
Watsen, K., "NETCONF Call Home (work in progress)", draft-ieft-netconf-restconf-10 (work in progress), December 2015, <https://tools.ietf.org/html/draft-ietf-netconf-call-home-17>.

[draft-ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ieft-netconf-restconf-10 (work in progress), 2016, <https://tools.ietf.org/html/draft-ietf-netconf-restconf-10>.

[draft-ietf-netconf-server-model]
Watsen, K., "NETCONF Server Model (work in progress)", draft-ieft-netconf-server-model-09 (work in progress), March 2016, <http://tools.ietf.org/html/draft-ietf-netconf-call-home-17>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, DOI 10.17487/RFC2315, March 1998, <http://www.rfc-editor.org/info/rfc2315>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <http://www.rfc-editor.org/info/rfc2782>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <http://www.rfc-editor.org/info/rfc6125>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <http://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <http://www.rfc-editor.org/info/rfc6763>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[Std-802.1AR-2009] IEEE SA-Standards Board, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", December 2009, <http://standards.ieee.org/findstds/standard/802.1AR-2009.html>.

12.2. Informative References

[draft-pritikin-anima-bootstrapping-keyinfra]
Pritikin, M., Behringer, M., and S. Bjarnason, "Bootstrapping Key Infrastructures", draft-pritikin-anima-bootstrapping-keyinfra-03 (work in progress), 2016, <https://tools.ietf.org/html/draft-pritikin-anima-bootstrapping-keyinfra>.

[RFC2939] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", BCP 43, RFC 2939, DOI 10.17487/RFC2939, September 2000, <http://www.rfc-editor.org/info/rfc2939>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <http://www.rfc-editor.org/info/rfc6698>.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <http://www.rfc-editor.org/info/rfc7317>.

Appendix A. Examples

A.1. Ownership Voucher

Following describes an example data-model for an ownership voucher. Real vouchers are expected to be encoded in a manufacturer-specific format outside the of scope for this draft.

A tree diagram describing an ownership voucher:

```
module: ietf-zerotouch-ownership-voucher
   +--rw voucher
      +--rw owner-id      string
      +--rw unique-id*    string
      +--rw created-on    yang:date-and-time
      +--rw expires-on?   yang:date-and-time
      +--rw signature     string
```

The YANG module for this example voucher:

<CODE BEGINS> file "ietf-zerotouch-ownership-voucher@2016-03-16.yang"

```
module ietf-zerotouch-ownership-voucher {

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-zerotouch-ownership-voucher";
  prefix "ztov";

  import ietf-yang-types { prefix yang; }

  organization
   "IETF NETCONF (Network Configuration) Working Group";

  contact
   "WG Web:   <http://tools.ietf.org/wg/netconf/>
    WG List:  <mailto:netconf@ietf.org>
    WG Chair: Mehmet Ersue
              <mailto:mehmet.ersue@nsn.com>
    WG Chair: Mahesh Jethanandani
              <mailto:mjethanandani@gmail.com>
    Editor:   Kent Watsen
              <mailto:kwatsen@juniper.net>";


  description
   "This module defines the format for a ZeroTouch ownership voucher,
    which is produced by Vendors, relayed by Bootstrap Servers, and
    consumed by devices.  The purpose of the voucher is to enable a
```

```
    device to ascertain the identity of its rightful owner, as
    certified by its Vendor.

    Copyright (c) 2014 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD
    License set forth in Section 4.c of the IETF Trust's
    Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see
    the RFC itself for full legal notices.";

  revision "2016-03-16" {
    description
     "Initial version";
    reference
     "RFC XXXX: Zero Touch Provisioning for NETCONF Call Home";
  }

  // top-level container
  container voucher {
    description
      "A voucher, containing the owner's identifier, a list of
       device's unique identifiers, information on when the
       voucher was created, when it might expire, and the
       vendor's signature over the above values.";
    leaf owner-id {
      type string;
      mandatory true;
      description
        "A Vendor-assigned value for the rightful owner of the
         devices enumerated by this voucher.  The owner-id value
         must match the value in the owner-certificate below";
    }
    leaf-list unique-id {
      type string;
      min-elements 1;
      description
        "The unique identifier (e.g., serial-number) for a device.
         The value must match the value in the device's IDevID
         certificate.  A device uses this value to determine if
         the voucher applies to it.";
    }
    leaf created-on {
```

```
        type yang:date-and-time;
        mandatory true;
        description
          "The date this voucher was created";
      }
      leaf expires-on {
        type yang:date-and-time;
        description
          "The date this voucher expires, if at all.  Use of this
           value requires that the device has access to a trusted
           real time clock";
      }
      leaf signature {
        type string;
        mandatory true;
        description
          "The signature over the concatenation of all the previous
           values";
      }
    }
  }
```

<CODE ENDS>

Appendix B. Change Log

B.1. ID to 00

- Major structural update; the essence is the same. Most every section was rewritten to some degree.
- Added a Use Cases section
- Added diagrams for "Actors and Roles" and "NMS Precondition" sections, and greatly improved the "Device Boot Sequence" diagram
- Removed support for physical presence or any ability for configlets to not be signed.
- Defined the Zero Touch Information DHCP option
- Added an ability for devices to also download images from configuration servers
- Added an ability for configlets to be encrypted

- Now configuration servers only have to support HTTP/S - no other schemes possible

B.2. 00 to 01

- Added boot-image and validate-owner annotations to the "Actors and Roles" diagram.
- Fixed 2nd paragraph in section 7.1 to reflect current use of anyxml.
- Added encrypted and signed-encrypted examples
- Replaced YANG module with XSD schema
- Added IANA request for the Zero Touch Information DHCP Option
- Added IANA request for media types for boot-image and configuration

B.3. 01 to 02

- Replaced the need for a configuration signer with the ability for each NMS to be able to sign its own configurations, using manufacturer signed ownership vouchers and owner certificates.
- Renamed configuration server to bootstrap server, a more representative name given the information devices download from it.
- Replaced the concept of a configlet by defining a southbound interface for the bootstrap server using YANG.
- Removed the IANA request for the boot-image and configuration media types

B.4. 02 to 03

- Minor update, mostly just to add an Editor's Note to show how this draft might integrate with the draft-pritikin-anima-bootstrapping-keyinfra.

B.5. 03 to 04

- Major update formally introducing unsigned data and support for Internet-based redirect servers.
- Added many terms to Terminology section.

- Added all new "Guiding Principles" section.
- Added all new "Sources for Bootstrapping Data" section.
- Rewrote the "Interactions" section and renamed it "Workflow Overview".

B.6. 04 to 05

- Semi-major update, refactoring the document into more logical parts
- Created new section for information types
- Added support for DNS servers
- Now allows provisional TLS connections
- Bootstrapping data now supports scripts
- Device Details section overhauled
- Security Considerations expanded
- Filled in enumerations for notification types

B.7. 05 to 06

- Minor update
- Added many Normative and Informative references.
- Added new section Other Considerations.

B.8. 06 to 07

- Minor update
- Added an Editorial Note section for RFC Editor.
- Updated the IANA Considerations section.

Authors' Addresses

Kent Watsen
Juniper Networks

EMail: kwatsen@juniper.net

Mikael Abrahamsson
T-Systems

EMail: "mikael.abrahamsson@t-systems.se