Zero Touch Provisioning for NETCONF/RESTCONF Call Home

draft-ietf-netconf-zerotouch-07

NETCONF WG

IETF 95 Buenos Aires

Recap

- At IETF 94, large reviews were received from ANIMA draft editor Max Pritikin and from co-author Mikael Abrahamsson.

- At the same time, an operator requested an ability for the bootstrapping data to optionally include a vendor-specific script.

Updates Since IETF 94

- Refactored draft into more logical sections
- Created new section for information types
- Added support for bootstrapping off DNS (discussed in upcoming slide)
- Added support for provisional TLS connections (discussed in upcoming slide)
- Bootstrapping data now supports scripts (discussed in upcoming slide)
- Device Details section overhauled
- Security Considerations expanded
- Filled in enumerations for notification types
- Added many Normative and Informative references
- Added new section Other Considerations
- Added a top-level Editorial Note section for RFC Editor
- Updated the IANA Considerations section

Bootstrapping Off DNS

- DNS-based service discovery (DNS-SD) [RFC6763]
  - Multicast: search for "_zerotouch._tcp.local."
  - Normal: search for "_zerotouch._tcp.example.com”
- Signed or unsigned redirect information can be returned
  - However, returning signed data is limited due to size constraints and multi-vendor applicability (i.e., ownership voucher format)
- Mapping fields to DNS records:
  - Address and port → DNS SRV record (unsigned data)
  - Everything else → DNS TXT records (signed data)

Provisional TLS Connections

- RESTCONF requires TLS
- However when device is provided unsigned redirect information, it cannot auth the server's TLS cert
- Thus the draft allows the device to blindly-accept the TLS cert, but then require:
  - That the bootstrapping data is signed
  - That the client doesn't provide any client auth
  - That the client doesn't send any notifications

Support for Scripts

- To support some addition pre-provisioning not covered by configuration (e.g., install FM/PM clients)

```
+--ro bootstrap-information
   +--ro boot-image
   |       ...
   +--ro configuration        anydata
   +--ro script?              string    // binary?
```

Change in Bootstrap Process

1. Check if running boot-image satisfies specified criteria
   - If not, then download, install, and reboot
2. Commit configuration into running config
3. If script, execute and process exit status code + stdout/stderr
4. Send notification (e.g., bootstrap-complete)

Support for Scripts (cont.)

- The script is a vendor-specific format
- Script should be executed with ‘root’ level permissions
- Script must provide exit status code as follows:

Code	Semantic	Next Step
0	script succeeded	continue
> 0	script reports having soft failure(s)	continue
< 0	script reports having a hard error	abort!

Open Issues

1. Ownership Voucher – formally define?
2. How to commit config? Merge/replace?
3. Signature over YANG data?
4. Removable storage details?

Let's discuss now…

#1: Ownership Voucher – formally define?

- Current ownership voucher is defined as being a *vendor-specific format*
- However:
  - A normative definition would fix the DNS multi-vendor issue
  - ANIMA team expressed interest in referencing it
    - Would need to add field to kind of ownership verification
      - e.g., absolute vs. logged-only
- If done, then we should also probably formally define the VRL (vendor revocation list) format as well…
- Thoughts?

#2: How to commit config? Merge/replace?

- Current text just says that the device commits the configuration, without clarifying how - should the device merge or replace the config?
- Considerations:
  - "Merge" can't remove default configuration, if needed
  - "Replace" requires all desired default configuration to be provided again
- Options:
  - Hardcode "replace" // always works, but large in size
  - Use a top-level flag // let deployments decide
  - Use edit-config or yang-patch // is this much granularity needed?
- Thoughts?

#3: Signature over YANG data?

Current text says that the signature is over the data in whatever form it's provided (XML or JSON)

This works great for all sources of bootstrapping data but, for the bootstrap server, requires that both the northbound application and the device access specific URL resources:

- /ietf-zerotouch-bootstrap-server:devices/device=123456/redirect-information
- /ietf-zerotouch-bootstrap-server:devices/device=123456/bootstrap-information

But this approach has issues:

- Server MUST NOT change its encoding in any way between the time the data was saved and when the device retrieves the data
- These two resources are under a choice node, and the device doesn't know up front which will be present.
- It is desirable for device to just fetch the top-level "device" resource, to get everything in one request

Options:

1. Keep as is (e.g., force device to try both URL resources) // up to 3 round trips instead of just one
2. Define a canonical format for XML and JSON encodings // Not sure if this is even possible…
3. Define an encoding-independent signature algorithm // tricky to define, not easy to implement
4. Encode both resources in a leaf having type "string" // even though it's YANG-encoded data

Thoughts?

#4: Removable storage details?

Current text says:

> Details such as the format of file system and the naming of the files are left to the device's manufacturer to define.

But an on list review recommends that the draft be more normative about this.

Opinions?

Final Stretch

- This draft is nearly done
  - Operational experience shows this to be true
  - IMO, we only need to address the open issues
- But would like more complete reviews to be sure!
  - Address open issues and then Last Call (i.e. ASAP)?
  - Solicit more big reviews and then decide?

Comments / Questions?