

# Zero Touch Provisioning for NETCONF/RESTCONF Call Home

draft-ietf-netconf-zerotouch-09

NETCONF WG  
IETF 96 (Berlin)

# Recap

- At IETF 95, we reviewed a significantly updated draft and its 4 open issues.
  - 2 issues now need to be verified
  - 2 issues remain open
  - 3 new issues added
- Very little on-list discussion...

# Updates Since IETF 95

- Added in missing "Signature" artifact example.
- Added recommendation for manufacturers to use interoperable formats and file naming conventions for removable storage devices.
- Added configuration-handling leaf to guide if config should be merged, replaced, or processed like an edit-config/yang-patch document.
- Added a pre-configuration script, in addition to the post-configuration script from -05.

# Open Issues

- #11: Ownership Voucher – formally define?
- #12: How to commit config? Merge/replace?
- #13: Signature over YANG data?
- #14: Removable storage details?
- #15: Enhanced script support
- #16: How to encode a chain of certs?
- #17: How to verify boot image integrity?

Discussed on following slides...


# #11: 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
- Very little update on this open issue.
  - Started working on a standard format with ANIMA team
- This issue needs to be taken to the list // too complex!

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

- Discussion so far has led us to this change:

```
+--ro bootstrap-information
  +--ro boot-image
  | ...
  +--ro configuration-handling? enumeration
  +--ro configuration? anydata
  +--ro script? script
```



merge  
replace  
edit-config  
yang-patch

None would  
be optional to  
implement!

- The enum not only specifies how to process the configuration, but it also specifies the format of the configuration
  - e.g., if merge/replace then raw config, else a specific format
- Questions:
  - Good idea?
  - Only Remove “edit-config”, since only good for XML and not “anydata” compatible?
  - Any other suggestions?

# #13: Signature over YANG data?

Current text says that the signature is over the data in whatever form it's provided (XML or JSON). This doesn't work well for bootstrap servers (i.e. a RESTCONF API), as how it is provided may vary (device could ask for XML or JSON).

Questions:

1. Can we assume bootstrap server has the Owner Certificate private key, and therefore can dynamically-sign whatever encoding it hands out?
  - this assumption would keep open many options, and seems reasonable, but some many not like it
2. Should we define “bootstrap info” (and redirect-info) using some other syntax (e.g., ASN.1) and present it as a binary blob in the API?
  - this is another way to eliminate the XML/JSON encoding issues...
3. Should we define an encoding-independent signature algorithm?
  - e.g., for each node in a depth-first traversal, convert value to a string and append to buffer to be signed.

Any other thoughts?

# #14: 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.

To be clear, the text is referring to the removable storage device (e.g., USB flash drive), not the device itself.

Based on Juergen's comment, I added to the above:

However, in order to facilitate interoperability, it is RECOMMENDED devices support open/standards based filesystems and to have a file naming convention that is not likely to have collisions with files from other vendors.

Any opinions before we close this issue?



# #15: Enhanced script support

The current draft allows for a single script. However, on-going implementation discussions suggest that there should be both a pre-commit and a post-commit script...

E.g., the pre-commit script could be used to download VNF images used by the configuration, and the post-commit script can do any necessary clean-up.

```
+-ro bootstrap-information
  +-ro boot-image
  | ...
  +-ro pre-configuration-script?  script  (NEW)
  +-ro configuration?
  +-ro post-configuration-script?  script  (RENAMED)
```

Any thoughts before closing this issue?

# #16: How to encode a chain of certs?

(related to system-keychain#1)

The draft currently states that the owner-certificate is just a single certificate

But the owner certificate actually needs to be presented along with its chain of intermediate certificates leading up to the trust anchor certificate known to the manufacturer's devices.

Here are some options:

1. Use an ordered-by-user leaf-list of the X.509v3 structures encoded using DER
2. Use a PEM “file” containing multiple BEGIN/END tags
3. Use a PKCS#12 structure from RFC 7292
4. Use a choice around both a PKCS#12 and a PEM

OpenSSL can translate between PKCS#12 and PEM well enough.

Opinions?

# #17: How to verify boot image integrity?

The current draft hardcodes the use of both the MD5 and SHA1 hash algorithms for the purpose of enabling a device to verify a downloaded boot-image, in case it doesn't have an embedded signature.

But a recommendation came to try to use a digest format that is more generic and less obsolete.

Some options:

- do nothing, keep as md5 and sha1
- replace both with sha256 (still hardcoded, but current)
- use a format that encodes both the alg's name + its value
  - while only supporting one alg for now (e.g., sha256)

Any other ideas?

# Final Stretch

- This draft is nearly done:
  - Operational experience shows this to be true
  - We only need to address the open issues!
- Next Steps:
  - Address open issues and then Last Call (i.e. ASAP)?
  - Solicit more big reviews and then decide?

Comments / Questions?