NETCONF Server and RESTCONF Server Configuration Models

draft-ietf-netconf-server-model-09

NETCONF WG

IETF 95 Buenos Aires

Recap

- The keychain module was presented at SAAG 94
  - to solicit security experts opinions and reviews
- Sean Turner agreed to review it
  - he provided a verbal review over the course of a very long phone call
- Updates made to -09 primarily to address his concerns.

Updates since IETF 94

- Renamed module ietf-keychain to ietf-system-keychain to disambiguate from the routing area working group's keychain model.
- Added an action statement to ietf-system-keychain to load a private key.
- Added a notification statement to ietf-system-keychain to notify when a certificate is nearing expiration and beyond.
- Converted all binary types to use ASN.1 DER encoding.
- Filled in the Security Considerations section.
- Added a Design Considerations section.
- Extended the Editorial Note section.
- Added many Normative and Informative references.

Open Issues

1. Missing feature statements in keychain module
2. Key-usage in generate-private-key action
3. Factor out common data within NC and RC modules?
4. How complete do the ssh/tls-server models need to be?
5. Split this draft into several drafts?
6. How to address the semi-configurable aspects of the keychain model?

Let's discuss…

#1: missing feature statements

- The keychain module is missing many feature statements
  - especially around algorithms and the action statements
- There isn't much to discuss right now
  - just be aware that an update is coming…

#2: key-usage in generate-private-key action

The current keychain module has this leaf definition in the input for the ”generate-private-key” action statement:

```
leaf key-usage {
  type enumeration {
    enum signing    { description "signing"; }
    enum encryption { description "encryption"; }
    // unclear if these should be somehow more
    // specific or varied.
  }
}
```

The key-usage was added per a recommendation from Sean Turner. Unless anyone has ideas, I’ll reach out to Sean again…

#3: Factor out common data within NC and RC modules?

- The current ietf-netconf-server and ietf-restconf-server modules both have a number of “uses” statements, that results in the same config needing to be entered more than once:

```
# grep uses ietf-netconf-server.yang | sed 's/^ *//'
uses ss:listening-ssh-server-grouping {
uses ts:listening-tls-server-grouping {
uses cert-maps-grouping;
uses endpoints-container {
uses ss:non-listening-ssh-server-grouping;
uses endpoints-container {
uses ts:non-listening-tls-server-grouping {
uses cert-maps-grouping;
uses x509c2n:cert-to-name;
```

- E.g, consider a server that supports both netconf-tls and netconf-ch-tls:
  - trusted-ca-certs, trusted-client-certs, and cert-maps all need to be specified twice, though almost assuredly the same…

#4: How complete do the ssh/tls-server models need to be?

- The current draft defines a minimum subset of SSH/TLS server config.
  - It does not support many config knobs provided by various SSH/TLS server implementations
- This issue seems similar to a module that needs to supported many vendors
  - Do we use LCD and expect augmentations to fill in missing parts when needed?
  - Or make an effort to fill in more and use feature statements to enable unsupported parts to be left out?
- Thoughts?

#5: Split this draft into several drafts?

- This draft defines a number of modules that might be used by other future drafts
  - Already the ietf-syslog module was thinking to reference the keychain module, in order to define something list a "ietf-tls-client" module

- It seems odd that such drafts would have to reference an RFC called "NETCONF Server and RESTCONF Server Configuration Models"

Proposal #1:

draft-ietf-netconf-system-keychain
draft-ietf-netconf-ssh-client
draft-ietf-netconf-ssh-server
draft-ietf-netconf-tls-client
draft-ietf-netconf-tls-server
draft-ietf-restconf-tls-client
draft-ietf-restconf-tls-server

Proposal #2:

draft-ietf-netconf-system-keychain
draft-ietf-netconf-ssh-client-server
draft-ietf-netconf-tls-client-server
draft-ietf-restconf-tls-client-server

Any other ideas?

#6: How to address the semi-configurable aspects of the keychain model?

- This issue is currently being discussed on list.
- Still, does anyone want to say something about it now?

Next Steps

- Close open issues (potentially splitting into many drafts)
- Update Call Home reference implementation
  - https://github.com/Juniper/netconf-call-home
  - Warning: just netconf-ssh and netconf-ch-ssh
    - No TLS or RESTCONF (is this a problem?)
- We will likely discuss again at IETF 96

Comments / Questions?