A Keychain-based Configuration Model for NETCONF and RESTCONF Servers

draft-ietf-netconf-server-model-08

SAAG
IETF 94 Yokohama
Draft’s Objective

• To define a configuration model for devices to:
  - Listen for NETCONF / RESTCONF connections
  - Call home using NETCONF / RESTCONF

• Note:
  - RESTCONF is HTTPS only
  - NETCONF is SSH or TLS
First Go At It → FAIL!

Using distinct models for NETCONF and RESTCONF

But each needs to specify:
  • Private keys
  • Trust anchors
  • Pinned certificates

And this data was duplicated twice in the NETCONF model – once for SSH and again for TLS
Current Idea: A keychain-based model
The ietf-keychain
Module

module: ietf-keychain
  +--rw keychain
  |   +--rw private-keys
  |   |   +--rw private-key* [name]
  |   |   |   +--rw name          string
  |   |   |   +--ro algorithm?  enumeration
  |   |   |   +--ro key-length? uint32
  |   |   |   +--ro public-key? string
  |   |   +--rw certificates
  |   |   |   +--rw certificate* [name]
  |   |   |   |   +--rw name     string
  |   |   |   |   +--ro chain? binary
  |   +--rw trusted-certificates* [name]
  |       +--rw name           string
  |       +--rw trusted-certificate* [name]
  |           +--rw name     string
  |           +--ro chain?   binary
  +--rw trusted-certificates
  |   +--rw name                   string
  |   +--rw trusted-certificate* [name]
  |       +--rw name           string
  |       +--ro chain?   binary

rpcs:
  +---x generate-private-key
  |   +---w input
  |   |   +---w name          string
  |   |   +---w algorithm     enumeration
  |   |   +---w key-length    uint32
  +---x generate-certificate-signing-request
  |   +---w input
  |       +---w private-key?     -> /keychain/private-keys/private-key/name
  |       +---w subject        binary
  |       +---w attributes?    binary
  +---ro output
  |       +--ro certificate-signing-request binary
<keychain xmlns="urn:ietf:params:xml:ns:yang:ietf-keychain">

<!-- private keys and associated certificates -->
<private-keys>
  <private-key>
    <name>ex-key-sect571r1</name>
    <algorithm>sect571r1</algorithm>
    <public-key>
      cztvaWRoc2RmZ2tqaHNNkZmdramRzZnZzZGtmam5idnNvO2RmanZvO3NkZmJpdmhzZGZpbHVidjvc21kZmhidml1bHNkYmZ2aXNiZGZpYmhzZG87ZmJvO3NkZ25iO29pLmR6Zgo=
    </public-key>
  </private-key>
  <certificates>
    <certificate>
      <name>ex-key-sect571r1-cert</name>
      <data>
        ...
      </data>
    </certificate>
  </certificates>
</private-keys>

<!-- trusted netconf/restconf client certificates -->
<trusted-certificates>
  <name>explicitly-trusted-client-certs</name>
  <trusted-certificate>
    <name>George Jetson</name>
    <certificate>...</certificate>
  </trusted-certificate>
  <trusted-certificate>
    <name>Fred Flinstone</name>
    <certificate>...</certificate>
  </trusted-certificate>
</trusted-certificates>

<!-- trust anchors for netconf/restconf clients -->
<trusted-certificates>
  <name>deployment-specific-ca-certs</name>
  <trusted-certificate>
    <name>Example.com</name>
    <certificate>...</certificate>
  </trusted-certificate>
</trusted-certificates>

<!-- trust anchors for random HTTPS servers on Internet -->
<trusted-certificates>
  <name>common-ca-certs</name>
  <trusted-certificate>
    <name>Example.com</name>
    <certificate>...</certificate>
  </trusted-certificate>
</trusted-certificates>
</keychain>
The ietf-tls-server Module

module: ietf-tls-server
  +--rw tls-server
    +--rw certificates
      |   +--rw certificate* [name]
      |       +--rw name   -> /kc:keychain/private-keys/private-key/certificates/certificate/name
      +--rw client-auth
        +--rw trusted-ca-certs?   -> /kc:keychain/trusted-certificates/name
        +--rw trusted-client-certs? -> /kc:keychain/trusted-certificates/name
TLS Server Example

<tls-server xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-server">
  <certificates>
    <certificate>
      <ex-key-sect571r1-cert>leafref</ex-key-sect571r1-cert>
    </certificate>
  </certificates>
  <client-auth>
    <trusted-ca-certs>
      <deployment-specific-ca-certs>leafref</deployment-specific-ca-certs>
    </trusted-ca-certs>
    <trusted-client-certs>
      <explicitly-trusted-client-certs>leafref</explicitly-trusted-client-certs>
    </trusted-client-certs>
  </client-auth>
</tls-server>
The ietf-restconf-server Module

module: ietf-restconf-server-new
  +--rw restconf-server
    +--rw listen {tls-listen}?
      |   +--rw max-sessions?   uint16
      |   +--rw endpoint* [name]
      |     +--rw name    string
      |   +--rw (transport)
      |       +--:(tls)
      |          +--rw tls
      |          |     +--rw address?   inet:ip-address
      |          |     +--rw port?     inet:port-number
      |          |       +-- <ietf-tls-server grouping>
      |          |       +--rw cert-maps <augmented in>
    +--rw call-home {tls-call-home}?
      +--rw restconf-client* [name]
        +--rw name   string
        +--rw (transport)
          +--:(tls)
          |     +--rw tls
          |        +--rw endpoints
          |           |     +--rw endpoint* [name]
          |           |     |     +--rw name    string
          |           |     |     +--rw address    inet:host
          |           |     |     +--rw port?     inet:port-number
          |           |       +-- <ietf-tls-server grouping>
          |           |       +--rw cert-maps <augmented in>
    +--rw connection-type ...
    +--rw reconnect-strategy ...
RESTCONF Server Example

<restconf-server xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server">
  <listen>
    <endpoint>
      <name>netconf/ssh</name>
      <tls>
        <address>11.22.33.44</address>
        <ietf-tls-server data goes here>
          <plus additional cert-maps data from augmentation>
        </ietf-tls-server>
      </tls>
    </endpoint>
  </listen>
  <call-home>
    <restconf-client>
      <name>config-mgr</name>
      <tls>
        <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>
        <ietf-tls-server data goes here>
          <plus additional cert-maps data from augmentation>
        </ietf-tls-server>
      </tls>
    </restconf-client>
  </call-home>
</restconf-server>
We need help to complete this work!

Please reach out to me if you would like to help