TLS support for CBS - Migration Plan

JIRA Associated

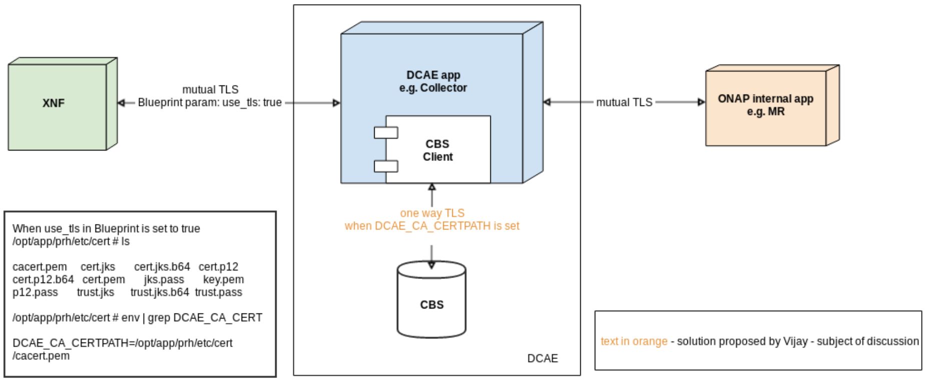
CBS HTTPS support **DCAEGEN2-1515** - Getting issue details... STATUS

1. CBS expose both secure/insecure + AAF cert **DCAEGEN2-1549** - Getting issue details... STATUS
2. Deployment update **DCAEGEN2-1550** - Getting issue details... STATUS
3. **SDK impact** (java - **DCAEGEN2-1552** - Getting issue details... STATUS / python - **DCAEGEN2-1551** - Getting issue details... STATUS)

Problem Statement :

CBS Api’s are used by all Service components to retrieve the configuration from consul during startup (and for periodic polling after). To support ONAP S3P security needs, Configbinding Service apis should be switched to HTTPS. As this has impact across all DCAE services, this has to be introduced in phased manner. El-Alto focus will on getting CBS HTTPS deployed and corresponding libraries updated.

Solution draft for review (author: Kornel Janiak):



XNF
mutual TLS
Blueprint param: use_tls: true
DCAE app
e.g. Collector
CBS
Client
mutual TLS
ONAP internal app
e.g. MR
one way TLS
when DCAE_CA_CERTPATH is set
CBS
DCAE
text in orange - solution proposed by Vijay - subject of discussion

```
When use_tls in Blueprint is set to true
/opt/app/prh/etc/cert # ls

cacert.pem    cert.jks      cert.jks.b64   cert.p12
cert.p12.b64  cert.pem      jks.pass       key.pem
p12.pass      trust.jks     trust.jks.b64  trust.pass

/opt/app/prh/etc/cert # env | grep DCAE_CA_CERT

DCAE_CA_CERTPATH=/opt/app/prh/etc/cert
/cacert.pem
```

Assumptions

- Not all service will switch to TLS interface for El-Alto
- CBS deployments must support both HTTPS and HTTP in-parallel
- SDK library (python and java) have separate api/version to let application choose migration
- ***Can*** deploy two instances in the same pod (CBS http and CBS HTTPS) under the same K8S service
- CBS is not be enabled for client-auth

Migration Plan

Following are impacts to components to be done in specified order

CBS Enhancement (DCAEGEN2-1549)

1. Support HTTPS enablement via environment variable
    - USE_HTTPS: set to “1” to use HTTPS, anything else is HTTP
    - HTTPS_KEY_PATH: path to the TLS private key
    - HTTPS_CERT_PATH: path to the TLS certificate
2. Use port 10443 if USE_HTTPS is set to “1”, otherwise port 10000

Deployment Enhancement (Helm chart updates) DCAEGEN2-1550

1. Modify existing **dcae-config-binding-service** charts to support the new environment and new CBS container version.
    a. Modify **values.yaml** service property to include two services: secure and insecure. Each service has the following properties:
        i. enabled - boolean flag to allow enabling and disabling the service. Allows running HTTP only, HTTPS only, or both. The default setting for El Alto is that both are enabled.
        ii. internalPort - the container port (10000 for HTTP, 10443 for https)
        iii. externalPort - the port used by other components inside the cluster to reach the service (10000 for HTTP, 10443 for https)
        iv. nodePort – the last two digits of the node port number (15 for HTTP, 14 for HTTPS)
    b. Modify **deployment.yaml** to deploy:
        i. if the secure service enable flag is set to true:
            1. the tls-init initContainer that sets stores certs in a volume in the pod
            2. a container with an instance of the config-binding-service image, with the USE_HTTPS, HTTPS_KEY_PATH, and HTTPS_CERT_PATH environment variables set to enable TLS
            3. a container with an instance of the filebeat logging forwarder, configured to read logs generated by the TLS-enabled instance of the config-binding-service
        ii. if the insecure service enable flag is set to true:
            1. a container with an instance of the config binding service image, with the USE_HTTPS environment variable set to 0, to enable non-TLS (HTTP) operation
            2. a container with an instance of the filebeat logging forwarder, configured to read logs generated by the non-TLS instance of the config-binding-service (this prevents problems that would be caused if both containers tried to write into the same logging volume)
    c. Modify **service.yam**l to deploy a Service named **config-binding-service** with port definitions based on the service property in the values. yaml.   The default service configuration will create a NodePort service with two ports, one for HTTP (inside the cluster on port 10000, outside the cluster on port 30415) and one for HTTPS (inside the cluster on port 10443, outside the cluster on port 30414).

K8s plugin updates (DCAEGEN2-1550)

1. Cloudify deployments of service components should include following environments
    - CONFIG_BINDING_SERVICE=<cbs_k8s_service_name>
    - DCAE_CA_CERTPATH=/opt/dcae/cacert/cacert.pem (this will be default unless overridden by component via blueprint)
    - CBS_CONFIG_URL=https://config-binding-service/service_component_all/<scn>
        - #scn Unique Servicecomponent name
2. Enable AAF cacert distribution (step to be done regardless of **tls_info** setting in blueprint) under DCAE_CA_CERTPATH

    Note: **tls_info** to be used for components supporting HTTPS as server. When specified, plugin will mount AAF certificate on application specific path specified. More details here - https://docs.onap.org/en/latest/submodules/dcaegen2.git/docs/sections/tls_enablement.html. In this case, DCAE_CA_CERTPATH will be overridden to use the path provided for exposing the cacert path)

    Below configuration is explicitly required in blueprint only when components required to support tls as server.

    ```
    tls_info:
      cert_directory: '<application path>'
      use_tls: true
    ```

Bootstrap pod (DCAEGEN2-1550)

1. Add new k8s plugin version including R4 version (1.4.13) in CM deployments
2. To keep existing components from breaking, continue to register “config-binding-service” and “config_binding_service” as services in Consul, with port 10000 as the service port.
3. Service registration on Consul will not be done for CBS TLS service. As components change to use TLS, they should use the Kubernetes DNS name (exposed via env CONFIG_BINDING_SERVICE) for the service along with port 10443.

Library Enhancement (CBS java sdk - DCAEGEN2-1552, CBS python util - DCAEGEN2-1551)

1. Verify if the new environment setting for TLS (below) added by K8s plugin is visible within POD.
    - CONFIG_BINDING_SERVICE=<cbs_k8s_service_name>
    - DCAE_CA_CERTPATH=<path>
2. If DCAE_CA_CERTPATH is defined, use the cacert for establishing secure end-point to interface with CBS (port 10443)
    a. An optional CBS_CONFIG_URL will be exposed providing the exact URL to be used for configuration retrieval. Application/Libraries can use this URL directly instead of constructing URL from HOSTNAME (which refers to ServiceComponentName) and CONFIG_BINDING_SERVICE env's. By default, this URL will use HTTPS CBS interface
3. If TLS env is undefined, use R4 service name and port (10000) to interface with CBS (HTTP)

Note: Libraries should stop using Consul service discovery to find CBS; instead rely on kubernetes DNS name (exposed via env CONFIG_BINDING_SERVICE) and port 10000 for HTTP and 10443 for HTTPS. Service registration on Consul will not be done for CBS TLS service

ServiceComponents (Optional for E release)

1. Switch to newer version of libraries (CBS SDK for java and python CBS utils)
    a. If not using library, component must use DCAE_CA_CERTPATH and 10443 for CBS HTTPS connection besides removing logic for Consul service discovery for CBS service.
    b. An optional CBS_CONFIG_URL will be exposed providing the exact URL to be used for configuration retrieval. Application/Libraries can use this URL directly instead of constructing URL from HOSTNAME (which refers to ServiceComponentName) and CONFIG_BINDING_SERVICE env's. By default, this URL will use HTTPS CBS interface
2. Update blueprint to use newer version of k8s plugin in blueprints (requires k8splugin version 1.4.13 or higher)

Discussion Notes

Updates from 10/17 discussion

Current implementation relies on trust.jks being available. Following options to be explored for SDK to interact with CBS HTTPS

- Option 1: Work/address issue around using cacert.pem for CBS connection (original proposal)
- Option 2: Enabled use_tls: true for all DCAE MS deployment (in blueprint) to ensure all AAF cert/trust and distributed (regardless of the MS /component being setup as server or not)
- Option 3: Modify K8s plugin to include trust.jks distribution by default along with cacert.pem

Current SDK change https://gerrit.onap.org/r/#/c/dcaegen2/services/sdk/+/94266/ relies on Option#2 and Piotr Wielebski reported issue on using cacert.pem

Updates from 10/24 discussion

2019-10-24 DCAE Meeting Notes

Option3 preferred; Damian/Nokia team will analyze the impact for k8s plugin updates.