Server Ssl

CB v4.2.5.150311.1434

March 11, 2015

Contents

Overview

Carbon Black makes heavy use of SSL Certificates for authentication and confidentiality between:

- your Carbon Black server and the Alliance server
- your Carbon Black server and the yum repository
- your sensors and your Carbon Black server
- your browser and your Carbon Black server

This document describes those communications, the location and configuration of each certificate and some Frequently Asked Questions on certificate management.

SSL Certificates

There are four different SSL certificates:

1. `/etc/cb/certs/alliance-client.crt` and `alliance-client.key`

   These certificates are installed with your in carbon-black-release RPM. They are unique per organization and valid for the lifetime of your relationship with Carbon Black. These certificates are used when your Carbon Black server communicates with the Alliance Server or `yum.carbonblack.com`.

2. `/etc/cb/certs/cb-client-ca.crt` and `cb-client-ca.key`

   This is the Certificate Authority used to issue client certificates to Cb sensors. These certificates are generated during `cbinit`, if and only if they do not already exist. This CA is used to sign sensor client certificates (see below) and authenticate sensor HTTP traffic. `nginx` validates any incoming sensor traffic was signed by this CA.

3. `/etc/cb/certs/cb-server.crt` and `cb-server.key`

   This is a normal HTTPS server certificate, used by nginx on tcp/443. Browser clients, REST API clients and sensors validate this certificate when connecting to the Cb server. This certificate is generated during cbinit, if and only if they do not exist. It is burned into each sensor at sensor download time.

4. In postgres, there are two classes of certificates: In the sensor_group table, there is one client certificate/key per sensor group. It is issued by the Certificate Authority in `cb-client-ca.crt`/`cb-client-ca.key` at group creation time. For the default group, this during cbinit. There is also a list of revoked certificates. Using `/usr/share/cb/cbssl` any sensor_group client certificate suspected of being compromised can be revoked with `cbssl`.

SSL Certificate Usage

- Cb server to Alliance:
  - Server: Cb Alliance server authenticates incoming connection based on the SSL Client certification from `alliance-client.crt/key`
  - Client: Carbon Black server authenticates Alliance certificate via normal root CA verification procedures
- Cb server to `yum.carbonblack.com`:
  - Server: `yum.carbonblack.com` authenticates incoming connections based on the SSL client certificate from `alliance-client.crt/key`
  - Client: The yum client authenticates the Cb yum server via normal root CA verification procedures
- Browser / REST API clients to Cb server:
  - Server: Cb Enterprise sends `cb-server.crt` as his identity. By default it is self-signed.
  - Client: clients authenticate Cb Enterprise server via normnal root CA verification procedures. Since the certificate is self-signed, this fails by default.
- Sensor to Cb server:
  - server sends cb-server.crt as his identity. sensor has cb-server.crt embdedded at sensor download time and validates the two match.
  - at sensor download time, sensor has embedded client cert and key, issued by cb-client-ca. server validates sensor client cert was issued by cb-client-ca, the client cert has been issued before to a sensor and the client cert has not been revoked. If the client cert does not match the expected certificate for the sensor group, but is from the valid ca and is not revoked, the server sends (and the sensor replaces) the new, expected client certificate.

cbssl

This utility provides the tools for managing Carbon Black Enterprise Server's SSL certificates.

FAQ

- I want to upgrade my server SSL certificate to a real, signed one so the browser quits complaining.

  Replace cb-server.crt/key. Any already deployed sensors must also be upgraded/re-installed out of band.

- I want to point an already-deployed sensor to another Cb server

  Copy cb-server.crt/key and cb-client-ca.crt/key to the new Cb server. On the new server, there are two options:

  ```
  - run cbinit
  - run cbssl
  ```

  stop cb-enterprise, run cbinit and then start cb-enterprise. cbinit is required to re-issue the sensor client certificates from the new CA.

  Finally, change server_url in Group Settings on the old server to point to the new server.

  Any sensors downloaded from the new server with the old keys will need to be upgraded/re-installed out of band.

- I believe a sensor client certificated was compromised, and my Carbon Black server has a interface in the DMZ. How do I ensure an attacker cannot use the sensor client certificate to push malicious data, compromising the integrity of the data in Carbon Black?

  Use cbssl to revoke the certificate for the compromised sensor_group. The remaining sensors in the group must be upgraded/re-installed out of band.