KIP-12 Knox Gateway TLS Keystore and Alias Should be Configurable

Status

Current-Status: Implemented

Discussion thread:

JIRA: KNOX-1756

Motivation

The location of the keystore housing the Knox Gateway TLS certificate is hardcoded to <*calculated from configs*>/keystores/gateway.jks and the certificate alias is hardcoded to “gateway-identity”. This limits the ability for external management facilities to setup a custom TLS identity for the Knox Gateway. For example, a host-wide, CA-signed, certificate.

Knox has configuration hooks for the following (optional) properties

- **Home Directory** - GatewayConfig.getGatewayHomeDir()
  - Gateway-site property: GATEWAY_HOME
  - System property: GATEWAY_HOME
  - Environment variable: GATEWAY_HOME
- **Data Directory** - GatewayConfig.getGatewayDataDir()
  - System property: GATEWAY_DATA_HOME
  - Environment variable: GATEWAY_DATA_HOME
  - Gateway-site property: gateway.data.dir
  - Calculated: [Home Directory] + [Path Separator] + “data”
- **Security Directory** - GatewayConfig.getGatewaySecurityDir()
  - Gateway-site property: gateway.security.dir
  - Calculated: [Data Directory] + [Path Separator] + “security”

Note: the calculation for the home directory is inconsistent with the other directory calculations. This inconsistency may be confusing to users and thus should be fixed to be

- System property: GATEWAY_HOME
- Environment variable: GATEWAY_HOME
- Gateway-site property: gateway.home.dir

The path to the Knox Gateway TLS keystore is calculated as

```
[Security Directory] + [Path Separator] + “keystores” + [Path Separator] + “gateway.jks”
```

Design

To make it easier to use an externally provided TLS identity, the Knox Gateway should allow the TLS keystore file and alias names to be configurable. The following properties should be made available:

- **TLS Keystore File Path** - GatewayConfig.getIdentityKeystorePath()
  - Gateway-site property: gateway.tls.keystore.path
  - Calculated: [Security Directory] + [Path Separator] + "keystores" [Path Separator] "gateway.jks"
- **TLS Keystore Password Alias** - GatewayConfig.getIdentityKeystorePasswordAlias()
  - Gateway-site property: gateway.tls.keystore.password.alias
  - Calculated: "gateway-identity-keystore-password"
- **TLS Keystore Type** - GatewayConfig.getIdentityKeystoreType()
  - Gateway-site property: gateway.tls.keystore.type
  - Calculated: :”jks”
- **TLS Key Alias** - GatewayConfig.getIdentityKeyAlias()
  - Gateway-site property: gateway.tls.key.alias
  - Calculated: “gateway-identity”
- **TLS Key Passphrase Alias** - GatewayConfig.getIdentityKeyPassphraseAlias()
  - Gateway-site property: gateway.tls.key.passphrase.alias
  - Calculated: "gateway-identity-passphrase"

Additional methods for GatewayConfig should be added to improve consistency and prepare for potential changes to signing key related configurations

- **Keystore Directory** - GatewayConfig.getKeystoreDir()

        - Calculated: [Security Directory] + [Path Separator] + “keystores”

- **Signing Keystore File Path** - GatewayConfig.getSigningKeystorePath()
    - Calculated: If gateway.signing.keystore.name is set, [Keystore Directory] + [Path Separator] + [Signing Keystore Name]; else [TLS Keystore File Path]

- **Signing Keystore File Type** - GatewayConfig.getSigningKeystoreType()
    - Calculated: If gateway.signing.keystore.name is set, value of gateway.signing.keystore.type (default: "jks"); else [TLS Keystore File Type]

- **Signing Keystore Password Alias** - GatewayConfig.getSigningKeystorePasswordAlias()
    - Calculated: If gateway.signing.keystore.name is set, value of gateway.signing.keystore.password.alias (default: "signing.keystore.password"); else [TLS Keystore Password Alias]

- **Signing Key Alias** - GatewayConfig.getSigningKeyAlias()
    - Calculated: If gateway.signing.keystore.name is set, value of gateway.signing.key.alias (default: "gateway-identity"); else [TLS Key Alias]

- **Signing Key Passphrase Alias** - GatewayConfig.getSigningKeyPassphraseAlias()
    - Calculated: If gateway.signing.keystore.name is set, value of gateway.signing.key.passphrase.alias (default: "signing.key.passphrase"); else [TLS Key Passphrase Alias]

Note: the *calculated* values are set so they are backwards compatible with older versions of Knox to ease the upgrading process.

Hardcoded values related to the identity and signing keystore location and relevant alias names should be removed, using the GatewayConfig methods to provide configured or calculated values. Assumptions that the keystore and key passwords are the master secret should be removed; however, if a password cannot be found via the AliasService, the fallback value should the master secret.