Fortanix®

# User Guide

## FORTANIX DATA SECURITY MANAGER – AUTHENTICATION & AUTHORIZATION

*VERSION 4.0*

Fortanix Inc. | 3910 Freedom Circle | Suite 104 | Santa Clara, CA 95052 | United States of America | ☎ +1 (650) 943-2484 |
✉ info@fortanix.com | 🌐 www.fortanix.com

## TABLE OF CONTENTS

# 1.0 INTRODUCTION

Welcome to the Fortanix Data Security Manager (DSM) Authentication and Authorization User Guide. This document describes the authentication and authorization mechanisms of the Fortanix DSM. It also contains the information related to:

- Planned enhancements for these mechanisms.
- Steps the users can take to safeguard their own authentication credentials.

## 1.1 FORTANIX DSM AUTHENTICATION AND AUTHORIZATION

Fortanix DSM provides access to its functions and APIs to two types of entities – humans (users), and machines (applications). There are many ways to authenticate to Fortanix DSM for both users and applications, which vary in terms of ease of use, integration with existing enterprise IAM (Identity and Access Management Systems), and level of security. Once authenticated, there is an elaborate access control mechanism which controls which entity has authorization to perform which function under what conditions.

# 2.0 DEFINITIONS

- **Fortanix Data Security Manager** -

  Fortanix DSM is the cloud solution secured with Intel® SGX. With Fortanix DSM, you can securely generate, store, and use cryptographic keys and certificates, as well as secrets, such as passwords, API keys, tokens, or any blob of data.

- **Accounts** -

  A Fortanix DSM account is the top-level container for security objects managed by the Fortanix DSM. An account is generally associated with an organization, rather than an individual. Security objects, groups, and applications belong to exactly one account. Different accounts are fully isolated from each other. *See support for more information.*

- **Users** -

  Users are associated with an email address. A user can be a member of one or more accounts. Depending on permissions, users can:

  - Perform management operations like adding or modifying users or groups

- Create security objects
- Change properties of security objects
- Review logs of Fortanix DSM activity

⚠ **Users cannot perform cryptographic operations. Only applications can perform cryptographic operations.**

- **Groups** -

  A group is a collection of security objects created by and accessible by users and applications which belong to the group. The user who creates a group automatically gets assigned the role of the Group Administrator. You can add more users to the group in the role of administrators or auditors. You can also add applications to the group to enable the applications to create and use security objects in that group. *See support for more information.*

  Access policies are set at the group level, so all security objects in a group share the same access policy. Any number of users and/or applications can be assigned to a group. *Some examples of usage of groups are given in the Authorization section*.

  Quorum policies can also be set at group level. A Quorum policy mandates that all security sensitive operations in that group would require a quorum approval. Such operations include using a key for cryptographic operations or deleting or updating a group. *See Quorum Policy for more information.*

- **Applications** -

  An application can use Fortanix DSM to generate, store, and use security objects, such as cryptographic keys, certificates, or an arbitrary secret. Applications can authenticate to Fortanix DSM using an API key (a secret token) or a TLS client certificate. An application can interact with Fortanix DSM using the REST APIs or using the PKCS#11, JCE, or CNG providers. *See support for more information.*

- **Security Objects** –

A security object is any datum stored in Fortanix DSM (for example a key, a certificate, a password, or other security objects). Each security object is assigned to exactly one group. users and applications assigned to the group have permission to see the security object and to perform operations on it. *See support for more information.*

# 3.0 AUTHENTICATION

## 3.1 USER AUTHENTICATION USING PASSWORD

The following authentication type is supported for users using password:

- **Username and password stored in Fortanix Data Security Manager**: This is done using the "log in without SSO" option.
  1. In the Fortanix DSM login screen, select the option "**LOG IN WITHOUT SSO**".
  2. Enter your password, and then click **LOG IN**.

## 3.2 USER AUTHENTICATION USING SSO – CONFIGURATION

Fortanix DSM accounts can be integrated with third-party Single Sign-On (SSO) providers. When an account is configured for SSO, users for that account will be able to login with their SSO credentials.

To set up SSO for your account:

1. Log in as an Administrator and click the **Settings** icon in the Fortanix DSM UI, and then click the **AUTHENTICATION** tab in the Account Settings page.
2. Select **SINGLE SIGN-ON**, and then add the desired SSO mechanism and provide the required configuration values.

**Warning:** Administrator lock-out: If the SSO mechanism is mis-configured, the Account Administrator created locally on Fortanix DSM will not be able to log in to the account. When updating the SSO configuration, ensure to select the **Account administrators can log in with password** option. This way, Account Administrators can continue to log in with password and access the account.

If you select either of the following options, then the **Mandatory two-factor authentication to log in with password** option is disabled.

- **All roles can log in with password**: Select this option during SSO configuration, if you want to allow any user role to log in to the Fortanix DSM account using their local password when the SSO mechanism is mis-configured.
- **No roles can log in with password:** Select this option during the SSO configuration, if you want no user role including the Administrator to log in to the Fortanix DSM account using their local password when the SSO mechanism is mis-configured.

Currently, the following SSO mechanisms are available for users:

- **SSO using a third-party identity provider**. The following protocols are supported:
  - **SAMLv2:**

    **Configuring a SAML provider:**

    To enable SAML for your account, first obtain the Identity Provider (IdP) metadata XML file. The IdP must meet the requirements set forth below. The SSO configuration page will inform you if the provided IdP metadata is compatible.

    **SAML Identity Provider Registration:**

    When configuring Fortanix DSM as a Service Provider with your IdP, provide the following information:

    - Entity ID: https://sdkms.fortanix.com/saml/metadata.xml
    - POST binding URL: https://sdkms.fortanix.com/saml

    **SAML Identity Provider Requirements:**

    To use a SAML IdP with Fortanix DSM, the IdP must:

    - Adhere to SAML 2.0, Web Browser SSO profile
    - Use one or more signing keys specified as an X.509 certificate
    - Use the `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` name format
    - Accept the POST binding for requests
    - Does not require signed requests
    - Use the POST binding for response
    - Sign responses, assertions, or both

To configure the user authentication using SAML, follow the below steps:

1. In the Authentication page, select **ADD SAML INTEGRATION** to configure SAML.
2. In the **Add SAML Integration** form, click **UPLOAD A FILE** to upload the configuration file (IdP metadata XML file), and then click **ADD INTEGRATION** to complete SAML configuration for user authentication.

- **OAuth / Open ID Connect:** To enable SSO using OpenID Connect / OAuth for your account, first obtain the following information from your Identity Provider (IdP):
  - Client ID
  - Client Secret

  **OAuth / Open ID Connect Identity Provider Registration:**

  You would need to register Fortanix DSM with your IdP to obtain these credentials. Provide the following values to your IdP:
  - Application type: web application
  - Redirect URL: https://sdkms.fortanix.com/oauth

  **Configure OAuth / Open ID Connect Identity Provider:**

  The IdP must meet the following requirements. To configure the IdP parameters in Fortanix DSM, the following information is required:
  - Provider name
  - Logo URL (optional)
  - Authorization endpoint URL
  - Token endpoint URL
  - Token endpoint authentication method (`client_secret_basic` or `client_secret_form`)
  - UserInfo endpoint URL (optional)
  - TLS configuration: use Global Root CAs or provide a custom CA certificate.
  - Prompt: select any of the following options for authentication request. By default, the **Consent** option is selected.

    **NOTE:** If you don't want any value for Prompt field, then deselect the Consent option.

    - **None:** select this option to initiate the silent authentication with the authentication request.

- **Login:** select this option to force a user authentication even if the user has been authenticated already with the authentication request.
- **Consent:** select this option to force prompting user consent with the authentication request.
- **select_account:** select this option to prompt the user to select a user account.

➢ Display: select any of the following options for Oauth server.

- **Page:** select this option to display the consent UI associated with a full user agent window.
- **Popup:** select this option to display the consent UI associated with a popup user agent window.
- **Touch:** select this option to display the consent UI associated with a device that leverages a touch user interface.
- **Wap:** select this option to display the consent UI associated with a feature-phone display type.

➢ Max Age: specifies the maximum amount of time that has elapsed in seconds since the OAuth provider last actively authenticated the end user.

**NOTE:** Ensure that the user knows about the supported OAuth parameters that he intends to use. As configuring the unsupported OAuth parameters might result in login errors. In case of such an error, only account administrator can log in to the account using the valid password only if "Only account administrators can log in with password" is selected at the time of account authentication configuration.

The user will be unable to log in to account if "No roles can login with password" is selected while configuring the single sign on (SSO) authentication.

Most of these parameters are published in a `.well-known` file provided by the identity providers. For example:

➢ Google

➢ Microsoft

**OpenID Connect / OAuth Identity Provider Requirements**

To use an OAuth / OpenID Connect IdP with Fortanix DSM, the IdP must:

- Support **Authorization Code Flow** described in OpenID Connect Core specification
- Support `email` scope
- Provide user's email address to Fortanix DSM in Token or UserInfo response
- Provide non-encrypted ID token during Token response

To configure user authentication using OAuth, follow the below steps:

1. In the Authentication page, click **ADD OAUTH INTEGRATION** to configure OAuth.
2. In the **Add OAuth Integration** form, add all the required details about the OAuth provider, and then click **ADD INTEGRATION** to complete OAuth configuration for user authentication.
   - LDAP: Fortanix DSM can be configured to authenticate users through an LDAP-compliant directory. Fortanix DSM supports the ldaps and ldap schemes. In both cases, the communication with the directory server is encrypted with TLS. When using the ldap scheme, the StartTLS operation is initiated immediately after connecting to the server.

     LDAP authentication is performed in two steps:
     - Resolve user's email address to a Distinguished Name (DN).
     - Authenticate to the directory using the DN and user-supplied password.

     **DN Resolution Methods**

     To resolve the user's email address to a DN, Fortanix DSM can be configured to use one of the following methods:
     - **Search the directory:** Fortanix DSM can search the directory to find the user object that matches the user's email address. The search is performed in a subtree and uses the following filter: `(&(objectClass={0})(mail={1}))` where `{0}` is the configured object class (for example, `User` or `inetOrgPerson`) and `{1}` is the user's email address. Some directories do not allow anonymous search, in which case a service account for Fortanix DSM should be created in the directory. When configured this way, the `mail` attribute must be set for user objects in the directory.
     - **Construct the DN from email address:** Given an email address of form name@domain, Fortanix DSM can be configured to lookup a format string based on the domain part and insert the name part in the format string to construct the DN. For

example, if example.com is configured with format string uid={},ou=users,dc=example,dc=com, then the email address test@example.com will be mapped to the following DN: uid=test,ou=users,dc=example,dc=com. The format string must include the placeholder {} which is replaced by the name part.

- **UPN login:** With Active Directory, Fortanix DSM can use the email address in place of the DN. When specifying an email address in place of the DN, Active Directory would check the value against the userPrincipalName attribute. If that attribute is not set, then Active Directory would accept values that match SamAccountName @ domain, where SamAccountName is the legacy user identifier attribute and domain is the fully qualified domain name of the Active Directory domain controller. We recommend setting the userPrincipalName attribute for all users in the directory when configuring Fortanix DSM with UPN login method.

To configure user authentication using LDAP, follow the below steps:

1. In the Authentication page, click **ADD LDAP INTEGRATION** to configure LDAP.
2. In the **Add LDAP Integration** form, add all the required details about the LDAP provider, and then click **ADD INTEGRATION** to complete LDAP configuration for user authentication.

## 3.3 CHANGING PASSWORD

To update the password, perform the following steps:

1. Log in to Fortanix DSM as a System Administrator.
2. To change the password, go to the **My profile** page.
3. Click the **Change Password** tab under password section.
4. Enter your current password in the **Current Password** field.
5. Enter your new password in **New Password** and **New Password (again)** field.
6. Click the **Change Password** button to confirm the changes.

The user password is updated successfully.

## 3.4 AUTHENTICATION USING 2FA (TWO FACTOR AUTHENTICATION) AT USER LEVEL

### 3.4.1 PREREQUISITES

- A Fortanix DSM user
- 2FA is currently supported using the FIDO2/WebAuthn standard, which is supported by devices such as YubiKey. This also works for FIDO U2F authenticators since FIDO2/WebAuthn is backwards compatible with them.

 **NOTE:**

- Your device must support FIDO2/WebAuthn to use 2FA.
  - Fortanix uses FIDO2/WebAuthn for the 2FA requirements.
  - Fortanix supports FIDO2 devices that provide either `packed` or `fido-u2f` attestation. *See https://www.w3.org/TR/webauthn-2/#sctn-defined-attestation-formats for more details*.
  - To use 2FA in Chrome and other browsers, the Fortanix DSM version must be greater than 4.8, and you must use the FIDO2 API endpoint `/sys/v1/session/auth/2fa/fido2` as Chrome and other browsers have deprecated the U2F API endpoint `/sys/v1/session/auth/2fa/u2f` since v98 https://developer.chrome.com/blog/deps-rems-95/#deprecate-u2f-api-cryptotoken.
- Only verified users can configure 2FA. A user is verified when they successfully re-verified their email ID. When the "**Sign-up email configuration**" in the **System Administration Settings** is set to **All Users** or **Users Since** the users will be given the option to re-verify their email address.

After user authentication using a password or user authentication using SSO is activated, an additional authentication layer can be added by enabling 2FA using a device that supports U2F or FIDO2/WebAuthn. To configure this, follow the steps below:

1. Click **My profile** to go to your profile settings.
2. In the option for **Two-step Authentication**, click **ENABLE** to enable two-factor authentication.

Once 2FA is enabled we will now have an additional authentication layer configured for either user authentication using password or user authentication using SSO which the user has already configured.

2FA can also be configured in the Fortanix DSM Quorum Policy for quorum approvals.

*For more information refer to the article https://support.fortanix.com/hc/en-us/articles/360016047771-User-s-Guide-Quorum-Policy.*

## 3.5 ACCEPTING TERMS & CONDITIONS

On the **My Profile** page, you must select the checkbox to accept the latest terms and conditions of Fortanix.


## 3.6 CONFIGURING 2FA AT ACCOUNT LEVEL

In *Section 3.3* we described how to configure 2FA at a user level using the User's profile. 2FA can also be configured at Fortanix DSM Account level. When 2FA is enabled at an account level, all users of the account will be required to set up 2FA before logging into the account.

### 3.6.1 CONFIGURING 2FA FOR AUTHENTICATION USING PASSWORD AT ACCOUNT LEVEL

1. Log in to Fortanix DSM as an Account Administrator and go to the **Settings** page for the account.
2. Select the **PASSWORD AUTHENTICATION** option.
3. Select the check box **Mandatory two-factor authentication for all team members**.
4. Click **SAVE CHANGES** to save the changes.
5. Once this setting is saved, all users of the account (including yourself) must configure 2FA using the "**My profile**" page as described in *Section 3.3*. This will add an additional layer of authentication for the account users. Without this configuration, they will not be able to log in to Fortanix DSM.

### 3.6.2 CONFIGURING 2FA FOR AUTHENTICATION USING SSO AT ACCOUNT LEVEL

1. Log in to Fortanix DSM as an Account Administrator and go to the **Settings** page for the account.
2. Select the **SINGLE SIGN-ON** option.
3. Configure the required SSO integrations using SAML, OAUTH or LDAP as described in *Section 3.2*.
4. Select the check box **Account administrators can log in with password**. This option prevents Account Administrators from being locked out of the account if the IdP is not configured correctly.
5. Now select the check box **Mandatory two-factor authentication to log in with password**.
6. Click **SAVE CHANGES** to save the changes.
7. Once this setting is saved, all users of the account (including yourself) have to configure 2FA using the "**My profile**" page as described in *Section 3.3*. This will add an additional layer of authentication for the account users. Without this configuration, they will not be able to log in to Fortanix DSM.

## 3.7 CONFIGURING 2FA AT THE SYSTEM LEVEL

In *Section 3.3* and *Section 3.4* we described how to configure 2FA at a user level and account level. 2FA can also be configured at a Fortanix DSM System level. When 2FA is enabled at a system level, all the System Administrators will be required to set up 2FA before logging into the System.

1. Log in to Fortanix DSM as a System Administrator.
2. To configure 2FA go to the **My profile** page and follow the steps described in *Section 3.3* to enable 2FA.
3. This will add an additional layer of authentication for the System Administrators. Without this configuration, they will not be able to log in to Fortanix DSM.

## 3.8 USER AUTHENTICATION USING SSO – USAGE

Once the configuration steps for user authentication using SSO are complete, the user can test the various authentication mechanisms using **LOG IN WITH SSO** option in the Fortanix DSM login screen. The user will now be presented with all the SSO authentication mechanisms that were configured for logging in to Fortanix DSM.

**Multiple Accounts:** Different accounts might have different SSO providers. A user can be in multiple accounts with different SSO providers. In these scenarios, the user will need to select which SSO provider to use during the login process. When switching accounts, a user might need to re-authenticate to satisfy the new account's authentication requirements.

## 3.9 APPLICATION AUTHENTICATION

Currently, there are five forms of authentication methods supported for applications:

- **Using a system generated API Key:** When you create an application in Fortanix DSM, an API key is used to authenticate the application. This API key is a random, secret token, that identifies an application in the same way as a password identifies a user. The user can copy this API key using the **COPY API KEY** button for the application.

Certain app integrations require username and password, so we use the **USERNAME/PASSWORD** tab for this requirement. This tab contains the **Username** and **Password** values. The **Username** (**app UUID**) is a unique identifier that the system generates for each application. The **Password** is the app secret that is also randomly generated by the system. For example, the Username and Password fields are used for the Fortanix DSM with VMware VSAN integration. Once this integration is set up, Fortanix DSM could be used for both vSphere VM encryption and VSAN encryption.

You can regenerate an app API Key such that old API key can continue to work for a configurable expiration period.

The following are the steps to configure the expiration period for an API Key:

1. Go to the detailed view of an app.
2. In the **INFO** tab, click **REGENERATE** in the **API Key** section.
3. In the **Regenerate API Key** dialog, set the expiration period using the **Expiration Setting** section and click **REGENERATE**.

- **Using a client TLS certificate:** Applications can also authenticate to Fortanix DSM using a TLS client certificate. To do this, select the **Certificate** option as the authentication method, and then upload a certificate using the **UPLOAD CERTIFICATE** button when you create a new

application. The user can also paste the certificate using the text box. We support certificates in PEM format only.

 **NOTE**: If an application needs to authenticate to Fortanix DSM using a certificate, then the App Id needs to be embedded in the certificate in one of the following ways:

1. **App Object Identifiers** (OID): Provided as the value of a custom OID in the certificate.
2. **Standard human-readable UUID encoding**: `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` provided as the value of CN.

- **Using a client TLS certificate issued by a trusted CA:** If your application's client TLS certificate is signed by a certificate authority, you can use this method to authenticate your application to Fortanix DSM.

  To use this method, you should specify the CA's certificate and the Subject Alternative Name (SAN) you expect in your application's certificate. The SANs supported are:

  - **DNS Name:** This is the default selection.
  - **IP Address**
  - **Directory Name:** This is a list of identifiers (key=value pairs). The user can add/remove any number of object identifiers for a Directory Name. The following object identifiers can be selected for directory names:
    - Common Name, Surname, Serial Number, Country Name, Locality, State/Province, Organization Name, Organizational Unit Name, Custom OID.

   **NOTE:** If you add multiple identifiers for Directory Name with the same key=value pairs, the value will be overwritten with the most recent value that is added for the key.

  When the app authenticates itself to Fortanix DSM using a client TLS certificate, it will verify that the app's certificate is signed by the specified trusted CA and that the SAN matches the expected value. To do this, select the **Trusted CA** option as the authentication method, Configure the SAN, and then upload a certificate using the **UPLOAD TRUSTED CA CERT**. The user can also paste the certificate using the text box provided. We support certificates in PEM format only.

  **Using Certificate Revocation List (CRL) in the Certificates:**

  A Certificate Revocation List is a digital record maintained by a Certificate Authority (CA) containing the serial numbers of digital certificates that have been revoked before their expiration date. The primary purpose of a CRL is to provide a mechanism for entities in a

cryptographic system to check the validity of digital certificates. By consulting the CRL, these entities can verify whether a particular certificate has been revoked due to data compromise or loss, thereby ensuring the security and integrity of digital communications.

Perform the following steps to use the CRL in the certificates:

1. **Generate a CRL:** Generate a CRL including the serial numbers of revoked certificates.
2. **DER Encode the CRL File:** Convert the generated CRL file to the DER format for compatibility with Fortanix DSM.
3. **Host the CRL File on HTTP:** Host the DER-encoded CRL file on a secure HTTP server accessible to the Fortanix DSM. Ensure that the server hosting the CRL file is reliable and secure.
4. **Configure DSM:** In the Fortanix DSM user interface, add a new app and select the **Trusted CA** as authentication method.
5. **Upload Root Certificate:**
    - Upload the root CA certificate.
    - Add client certificate Subject Alternative Name (SAN) details. Ensure that these details correspond to both revoked and non-revoked certificates.
    - Associate the CRL file hosted on the HTTPS server with the Trusted CA entry. This allows Fortanix DSM to fetch and verify the revocation status of client certificates.
6. Select the **Verify client certificate revocation status** check box to activate CRL verification during authentication. This option ensures that DSM verifies the revocation status of client certificates against the configured CRL. By default, this check box is not selected.

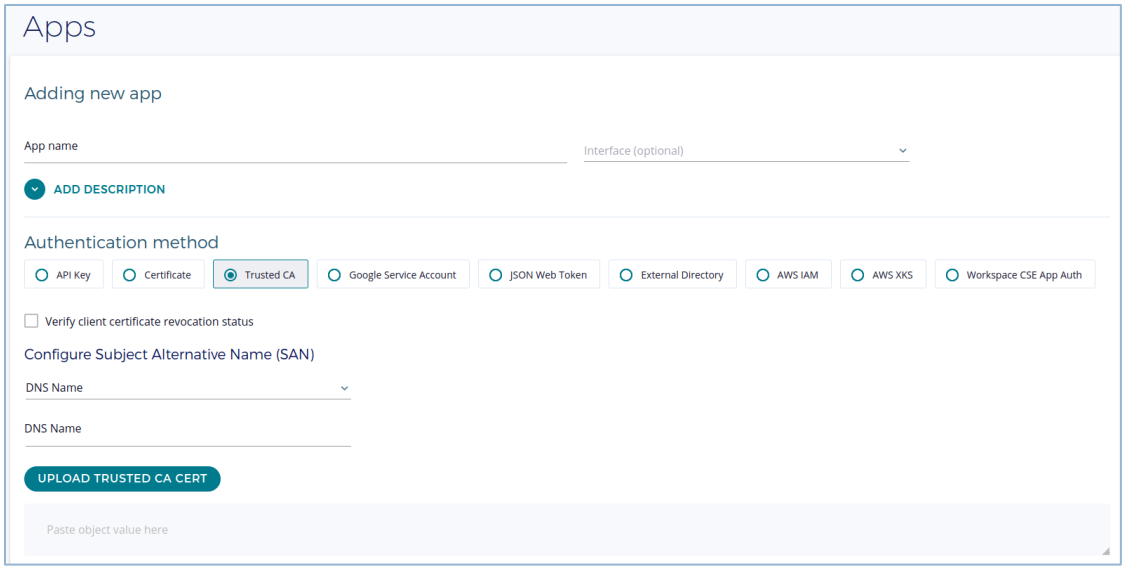


**FIGURE 1: SELECT THE CHECK BOX**

7. **Authentication:** Run the following command to authenticate the certificates using the DSM app:

```
curl https://apps.sdkms.test.fortanix.com/sys/v1/session/auth -X
POST -u <APP_UUID> --cert <PATH/TO/CERTIFICATE> --key
<PATH/TO/PRIVATE/KEY>
```

Where,

- `<APP_UUID>` refers to the UUID of the app.
- `<PATH/TO/CERTIFICATE>` refers to the path of client certificate.
- `<PATH/TO/PRIVATE/KEY>` refers to the path of client key.

- **Google Service Account** identifier, which is used by a service account in Google Cloud to use the external KMS interface from the GCP EKM interface. To learn more about this scheme, please refer to the article Using Fortanix Data Security Manager with Google Cloud EKM Interface.
- **JSON Web Token (JWT):** Applications can also authenticate to Fortanix DSM using signed JSON Web Tokens. A signed JWT is a cryptographically verifiable token that carries information about a subject and is encoded as three base64 encoded sections separated by dots. For example, the following:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjFhYWEwNWJjN2Q0NDQ1NWFmNzRmY
TdmMDBhZDRmMjgyMTQ2YTQ2NzMifQ .
```

eyJzdWIiOiJteSBhcHAiLCJpc3MiOiJodHRwczovL2V4YW1wbGUuY29tIiwiYXVkIjoiaHR0cHM6Ly9zZGttcy5mb3J0YW5peC5jb20iLCJpYXQiOjE1NDYzMzkwMjIsImV4cCI6MTU0NjM0OTAyMn0 .KUtIf6zJGWozRplp32Vt-vt-Sy1TyXmL5svWJHvdKJ1q20zrGNoUhcvGYOSF7X3mIvxKXC_qGNQJZjsXA5KAJRhp-6SIn8MsexLbnfioxDny7ZuPHPo22pCS55xPukxZQSWW6JLRk7ITHvrYqY4boDao7bxZdoPshuw2ekZ6UHS5GdaWPcN-od_xS0nYqhdiI4gw-A23IrneFwwVCfziQ-u_tNuqXL0SjT3UbYPbtfkcQEfBdJpKPyU3ZdJ_gAKNj071vvAMKwM53wXclu-w7NKyNfgA1zz-S2gQfy643e61Lg8i-mlabwK7hXEFCx5ksnTpff037BRDUnzNphvOjQ

is a signed JWT and it carries the following information in its first 2 base64 encoded parts:

Header:

```
{
 "alg": "RS256",
 "typ": "JWT",
 "kid": "1aaa05bc7d44455af74fa7f00ad4f282146a4673"
}
```

Body:

```
{
 "sub": "my app",
 "iss": "https://example.com",
 "aud": "https://sdkms.fortanix.com",
 "iat": 1546339022,
 "exp": 1546349022
}
```

The third part of the token is a cryptographic signature that can be independently verified by anyone with knowledge of the public key that was used to sign the token.

All the API endpoints accessible to apps also accept JWT authentication tokens if the app is configured with JWT authentication mechanism. An app holding a valid JWT token works

similarly to an app with a valid API key. The JWT token needs to go in the HTTP Authorization header, encoded as the password with the username being the account ID of the account where the app is created: `Authorization: Basic $TOKEN` where `$TOKEN` is a base64 encoded `$ACCT_ID:$JWT`.

The following snippet shows how to do this encoding in a shell script (BASH):

```
$ echo -n "$ACCT_ID:$JWT" | base64 -w0
```

**NOTE**:

1. The JWT body must include the following fields:
   - `sub` the subject of the token must match the app's name in Fortanix DSM.
   - `iss` the issuer of the token is validated against a list of issuers.
   - `aud` the audience of the token must include the Fortanix DSM endpoint URL, e.g., https://sdkms.fortanix.com this might be different for you if using an on-prem Fortanix DSM cluster).
2. In some scenarios, we also require the presence of `kid` field in the JWT header. This is discussed below in the section Signing Keys**.**

Fortanix DSM verifies the authentication request by (not necessarily in the order listed):

- Looking up the app in the specified account using the JWT subject field as app name,
- Looking up the public key for signature verification,
- Verifying the token's signature using that public key,
- Verifying temporal constraints including `nbf` (not before) and `exp` (expiration time) if present in the token,
- Verifying other constraints listed above (`iss` and `aud`).

To configure an app to use this authentication method, you would need to specify the following:

**Valid Issuers**

One or more valid issuers. This information is used to validate the `iss` field of the JWT.

### Signing Keys

Either **STORED SIGNING KEYS** or **FETCHED SIGNING KEYS**. This is used to validate the signature on the JWT:

- **Stored**: provide one or more public keys by uploading or pasting the key in PEM format. When specifying multiple keys, specify the Key ID so that keys can be differentiated by unique Key IDs. In this case, the signed JWT should also indicate which signing key was used to sign that token by specifying `kid` in its header section.
  Note that if the signed JWT includes the `kid` field in its header you should always specify Key ID for the public key(s) configured on the app regardless of the number of signing keys specified.
- **Fetched:** provide a URL that can be used to fetch the signing key(s). To improve performance, Fortanix DSM will cache the fetched keys, but you can control how long the keys are cached as. The specified URL should return the signing keys in one of the following formats:
  - JWK Set: *please refer to Section 5 of RFC 7517 for details.*
  - A JSON object mapping key IDs to base64 encoded public keys in DER format. Here is an example:

```
{
 "f00cd596-5108-4349-9dc2-0ce14c56b1f6": "MIIBoDANBgkqhkiG9w0BAQEFA
AOCAY0AMIIBiAKCAYEAo6f6tb41PfkLk69EREUjQLwDArTEdYqKd9+QjlLm89Sv7sFz
gtL/aNmzSPgJ9t1m0XOIMm51QkNEMA0tA9yEf3AjP6U/4F+f7A2jTLWY09wsXG3qulA
w5FC78xCoK6pBBIwev2LRzo0Blz1exAv9mOXP/fGATQOdnwDRRr6lAbnynR86qxfF/g
e0r+OVUEni4eOMAvr6lYrQeXKl1hQpb86JWq3eSdk/LYFB+8366bogjN+oLLb7LIBbm
EEctZ7ygIxnKiCIhZS3C0tTHrBi4aNTLps7UPlJna7UqmmN5bdhgih9SjjnP7CJnZG8
ZsB3JMXlpBE0dP5nnGW20ZqxbxBXYgSxOd1SnGIp/hI7aQctyB9M9di2C57gtlDB++t
sAZg9FWZo1y8IMY4e1AEYHxhq5PEVhBL3jW0xXrZrcuoB0c7i7aX0B4vRnvNdflZ9om
wP2zDd2jYcuqPaDkygjzuoDTsjcC/NvjHWmK1EEOvgsDDOMMXwQ9cErsEtLUbHAgED"
}
```

 **NOTE**: When using fetched signing keys, the signed JWT should always contain a kid field in its header.

We recommend using fetched signing keys when you need to periodically rotate the signing keys.

*Please refer to **RFC 7519** and **RFC 7515** for more information about JSON Web Tokens.*

- **External Directory:** In this method, Fortanix DSM extends the Application's authentication model to allow the application to authenticate using its credentials in Active Directory (AD). When this option is selected, the user is presented with a list of enrolled external directories and the user can choose a directory.

   **NOTE:** The user should name the application to have a full distinguished name of the application in AD.

  These types of applications authenticate to Fortanix DSM by providing the `app_id` and credentials of the app in AD. Fortanix DSM will look up the app by `app_id`, find the corresponding external directory and present the credential to it for authentication. If the external directory successfully authenticates, then Fortanix DSM will continue to find authorization for the app.

- **AWS IAM:** In this method, Fortanix DSM extends the Application's authentication model to allow the application to authenticate using AWS Identity and Access Management (IAM) that provides fine-grained access control across all of AWS.

   **NOTE:** The **App name** must match the AWS ARN associated with the calling entity.

Fortanix DSM Authenticates with AWS IAM using the Signature Version 4 signing process. Signature Version 4 (SigV4) is the process of adding authentication information to AWS API requests sent by HTTP. For security purpose, you must sign most requests to AWS with an access key. The access key consists of an access key ID and a secret access key, which are commonly referred to as your security credentials.

**How to Follow the Signature Version 4 Process**

1. Create a canonical request.

2. Use the canonical request and additional metadata to create a string for signing.
3. Derive a signing key from your AWS secret access key. Then use the signing key and the string from the *Step 2* to create a signature.
4. Add the resulting signature to the HTTP request in a header or as a query string parameter.

When an AWS service receives a request, it performs the steps that calculate the signature sent in the request. AWS then compares its calculated signature to the one you sent with the request. If the signatures match, the request is processed. If the signatures do not match, the request is denied.

You can change between app authentication methods such that the previous authentication method will continue to work for a configurable expiration period.

The following are the steps to configure the expiration period for the previous authentication method:

1. Go to the detailed view of an app.
2. In the **API Key** section, click the drop-down list for **Change authentication method** and select the new authentication method.
3. Click the **SAVE** button.
4. In the detailed view of the new authentication method page, set the expiration period in the **Expiration Setting** section and click **UPDATE.**

**References:**

- **Using Fortanix Data Security Manager with Google Cloud EKM Interface**: https://support.fortanix.com/hc/en-us/articles/360030816111-Using-SDKMS-with-Google-Cloud-EKM-Interface
- **RFC 7519**: https://tools.ietf.org/html/rfc7519
- **RFC 7515**: https://tools.ietf.org/html/rfc7515
- **RFC 7517**: https://tools.ietf.org/html/rfc7517

- **AWS XKS:** Select this option to protect the data in AWS with keys stored in Fortanix DSM that users can use to perform cryptographic operations. *For more details, refer to Cloud Key Management / BYOK documentation.*

- **Workspace CSE App Auth:** Select this option to add a Google Workspace CSE user as an app. Ensure that the app name of the CSE user is same as email address of an existing Google Workspace user. *For more details, refer to Integration Guide: Fortanix DSM with Google Workspace CSE.*

## 4.0 ENABLE OAUTH FOR AN APPLICATION

A user can configure OAuth for an app to authorize the app to perform cryptographic and key management operations on their behalf in groups that the user has an administrator role. To enable OAuth:

1. In the create application form, click the toggle **Enable OAuth**.
2. In the **Authorized Redirect URIs** field, enter the redirect URL.
3. Click **SAVE** to save the application.

**NOTE**: OAuth can also be enabled from the detailed view of an app as shown below:
Now, when the external application performs OAuth and requests permission to access the Fortanix DSM group to perform crypto and key management operations, the user will be redirected to the Fortanix DSM login page for authorization:

1. The user must log in to Fortanix DSM to authenticate and select the account that contains the app for which OAuth was enabled.
2. The user will see the following consent page.
4. The user must select a default group from the list of groups in which the user has an administrator role that will store the security objects created by the app.
5. Click **AUTHORIZE** to authorize the app.

*For more details refer to the OAuth authorization framework.*

## 5.0 CREATE APPLICATIONS AND GROUPS PROGRAMMATICALLY USING APP AUTHENTICATION METHODS

### 5.1 CREATE ADMINISTRATIVE APPS

Sometimes we might need to create groups and applications programmatically using application authentication methods instead of creating the same by logging into the Fortanix DSM UI. For example, this will help in onboarding users of a particular sub-department of an organization in a

more automated manner using their tools instead of human intervention with Fortanix DSM. This can be achieved using the Fortanix DSM Administrative Apps feature in the **Account Settings** page.

**NOTE**: The Fortanix DSM Administrative Apps can only be created by Account Admins.

To create an Administrative App:

1. Go to the Administrative Apps page and click **ADD ADMINISTRATIVE APP** button.
2. The Administrative App will display all the authentication methods that are currently supported by Fortanix DSM applications:
   - API Key
   - Certificate
   - Trusted CA
   - JWT Web Token

**NOTE**: There can be more than one administrative apps created with each administrative app mapped with different authentication methods.

3. Click **CREATE** once the authentication method is configured for the administrative app.
4. Once the administrative app is created, a user can use one of the authentication methods configured for an administrative app. For example: **COPY API KEY** to authenticate to Fortanix DSM programmatically.

**NOTE**:

- Administrative apps can create and delete quorum approval requests.
- Administrative apps can manage keys without the app manageable permission.

## 5.2 DISABLE/ENABLE ADMINISTRATIVE APP

To disable an administrative app:

1. Go to the Administrative Apps table view and click the toggle button to disable/enable an administrative app.

### 5.3 DELETE ADMINISTRATIVE APP

To delete an administrative app:

1. Go to the detailed view of the administrative app and click the **DELETE ADMINISTRATIVE APP** button at the bottom of the page.

### 5.4 VIEW AUDIT LOGS OF ADMINISTRATIVE APP

To view the audit logs of an administrative app:

1. Go to the detailed view of an administrative app, and click the **AUDIT LOG** tab.

### 5.5 IP WHITELISTING OF APPLICATION IN ADMINISTRATIVE APPS

*Refer to **Section 5.0** for more details.*

## 6.0 IP WHITELISTING OF APPLICATIONS

Whitelisting of IP address for an application allows an application to authenticate itself only if it comes from the whitelisted IP address. The IP whitelisting feature allows a range of IP addresses to authenticate and access a group.

**NOTE**:

- The IP Whitelisting of application is an additional restriction on authenticating an application among the available authentication methods present such as authenticating using API Key, Certificate, Trusted CA, Google Service Account, JSON Web Token, and AWS IAM.
- This authentication mechanism should not be relied upon for security since the IP address can be fabricated.
- If you do not see the IP Whitelisting setting enabled in the UI, please contact your System Administrator.

To specify the range of IP address for the application:

1. Go to the detailed view of an application and click the **EDIT** button to the right of **Allowed IP-addresses** field.
2. Select **Restrict authentication to trusted IPV4 CIDRs** option.
3. Enter the **CIDR.** Classless inter-domain routing (CIDR) is a set of Internet protocol (IP) standards that are used to create unique identifiers for networks and individual devices.
4. You can add multiple CIDRs using the **ADD ANOTHER CIDR** option.

5. Click **SAVE** to save the settings.

## 7.0 CONFIGURING LDAP AUTHENTICATION WHEN USING SSH

To configure LDAP authentication on the Fortanix DSM node while using SSH, follow the steps below:

1. Install the LDAP dependency packages using the command:

```
sudo apt-get install ldap-auth-client nscd ldap-utils
```

2. During the installation, you will be prompted to provide the following details of your LDAP server:
   - **LDAP Uniform Resource Identifier (URI)**: This can be LDAP Server's IP address or hostname.
   - **Domain Name (DN)** of the LDAP search base.
   - **LDAP version** to use.
3. Install the configuration package `ldap-config` using the command:

```
sudo apt install ldap-config
```

4. After the installation is complete, the details of the configuration provided in the previous steps can be verified in the file `/etc/ldap.conf`.
5. Verify the Pluggable Authentication Module (PAM) configuration at `/etc/pam.d/common-session` file.

   The file should show the following line:

```
session required        pam_mkhomedir.so skel=/etc/skel/ umask=0022
```

6. The configuration of LDAP authentication is complete. The connectivity can be verified using SSH with one of the users of the LDAP directory.

## 8.0 AUTHORIZATION

Once a user or an app is authenticated, it still needs the authorization to perform an operation on Fortanix DSM. Fortanix DSM provides fine-grained authorization controls that can broadly be categorized into "time-based authorization", "role-based access control (RBAC)", "quorum-based authorization", "key-based authorization", "LDAP authorization" and "authorization for plugins".

## 8.1 TIME-BASED AUTHORIZATION

When an application or a user authenticates to Fortanix DSM, a bearer token is granted which authorizes the user or the app to make API calls while the bearer token is valid. In the current implementation, the bearer token expires after a period of inactivity. A System administrator can configure this period. A user may also change the API key of an app at any time which invalidates any existing bearer tokens immediately.

## 8.2 ROLE-BASED AUTHORIZATION

Fortanix DSM provides four roles at an account level – an Account Administrator, an Auditor, an Account Member, or a Custom account role. There are three roles at the group level – a Group Administrator, a Group Auditor, or a Custom group role.

The Account Administrator, Auditor, Account Member, Group Administrator, and Group Auditor are called "**legacy roles**".

**Account Administrator** is a Group Administrator for all groups in the account. This role includes all account-level permissions as well as giving users the Group Administrator role in all groups in the account.

**Account Member** has the ability to create groups, act as quorum policy reviewer and key custodian, and read permissions at the account level. This role does not automatically entitle the user to any group. A user with this role can have selective access to any group if invited.

**Account Auditor** is a Group Auditor for all groups in the account. This role only entitles the user to read permissions (such as getting audit logs, and so on.) at the account level.

**Group Administrator** allows the user to do everything in a group.

**Group Auditor** allows read permissions in a group such as getting security objects.

At the system level, Fortanix DSM provides two more roles – a System Administrator, and a System Operator. They roughly map to an Administrator and an Auditor (read-only) role at the account level.

With the introduction of Custom user roles, it is possible to define a Custom user role with an arbitrary set of permissions. There are two types of Custom user roles: **Custom account role** and **Custom group role**. *For more details on how to create and manage Custom user roles, refer to User's Guide: Custom Roles.*

**NOTE**: Note that users in the System Administrators account can only have Account Administrator or Account Auditor roles and those users are called sysadmins and sysops respectively. The powers afforded to those users are separate from regular accounts. We may introduce Custom roles for System Administrators account users in the future.

Please see the following chart for the actions allowed for every role:

| ACTIONS | SYSTEM ADMIN | SYSTEM OPERATOR | ACCOUNT ADMIN | ACCOUNT MEMBER | ACCOUNT AUDITOR | APP | ADMIN APPS |
|---|---|---|---|---|---|---|---|
| **Add and manage applications** | | | **YES** | **YES** | | | **YES** |
| **Add and manage users, change user roles** | | | **YES** | | | | **YES** |
| **Create accounts** | | | **YES** | | | | |
| **Manage accounts** | | | **YES** | | | | **YES** |
| **Create and manage groups** | | | **YES** | **YES** | | | **YES** |
| **Cryptographic operations** | | | | | | **YES** | |
| **Create and manage plugins** | | | **YES** | **YES** | | | **YES** |
| **Invoke plugins** | | | **YES** | **YES** | | **YES** | **YES** |
| **Key management (add, edit, delete keys)** | | | **YES** | **YES** | | **YES** | **YES** |
| **Monitor Fortanix Data Security Manager** | **YES** | **YES** | | | | | |

| ACTIONS | SYSTEM ADMIN | SYSTEM OPERATOR | ACCOUNT ADMIN | ACCOUNT MEMBER | ACCOUNT AUDITOR | APP | ADMIN APPS |
|---|---|---|---|---|---|---|---|
| **Install and configure Fortanix Data Security Manager** | **YES** | | | | | | |
| **Upgrade software** | **YES** | | | | | | |
| **View audit logs** | | | **YES** | **YES** | **YES** | | **YES** |

TABLE 1: FORTANIX DATA SECURITY MANAGER USER ROLES AND ACTION

**NOTE for Account Member Role:**

: When we add a user with "Account Member" role and assign this user to a group as Group Admin or Group Auditor, then this user is allowed to do operations such as – **Add and manage applications**, **Create and manage groups**, **Create and manage plugins**, **Invoke plugins**, **Key management (add, edit, delete keys)**, and **View audit logs.**

: When we add a user with "Account Member" role without assigning the user to a group, then this user is allowed to do operations such as - **Create and manage groups**, **Key management (add, edit, delete keys)** for the keys inside the group created by the account member, and **View audit logs**.

## 8.3 QUORUM-BASED AUTHORIZATION

Fortanix DSM allows a quorum policy to be set on a group in an account, such that every security-sensitive operation in the group requires a quorum approval to be obtained.

- The quorum policy is defined as a conjunctive or disjunctive set of quorum groups defined in the form of (M of N approvers). It is the minimum number of approvals required among the total number of Group Administrators for the group.
- A policy may also include the specific identity of users who form the quorum, and not just the size of the quorum.
- An advanced policy could be a combination of quorum rules. For example, a quorum could be defined as “one out of users A and B”, and “three out of users C, D, E, F, and G”.

**Quorum Approval Workflow**

Any sensitive operation performed in the group (for example: using a key for cryptographic operations, modifying the quorum policy, deleting/updating a group and so on) triggers a quorum event where notification is sent to all the approvers in the quorum group and a workflow is triggered.

- This involves sending notification to all users who can grant approval. This is done by sending emails, as well as generating a task in the approver's accounts, which they see on the dashboard as soon as they login to their Fortanix DSM account.
- The users can then grant approvals from the UI. The sensitive operation is blocked until the quorum is met.
- Once a quorum is reached, the operation is performed, and the quorum approval is written into the audit logs including the names of users who approved the request.

Quorum-based authentication ensures that some high-value keys can be protected from misuse by a single rogue Administrator.

## 8.4 KEY BASED AUTHORIZATION

In Fortanix DSM when a security object (key) is created, there are some crypto operations that are permitted with the security object. These operations can be defined during the creation of a security object.

## 8.5 LDAP AUTHORIZATION FOR USERS

Fortanix DSM can also leverage group membership in an LDAP-compliant directory to dynamically assign users to groups. This requires mapping LDAP groups to Fortanix DSM groups. This is achieved by defining external roles in Fortanix DSM and mapping these external roles to Fortanix DSM groups. After a user authenticates to Fortanix DSM using LDAP, Fortanix DSM retrieves the list of directory groups that the user belongs to. If the retrieved groups map to Fortanix DSM groups, the user is added to Fortanix DSM groups for the current session.

### 8.5.1 DEFINING EXTERNAL ROLES

Account Administrators can create external roles for the account. To do that, they must have added one or more LDAP integrations in Account Authentication settings. Using the LDAP search functionality, Account Administrators can look for group objects in an LDAP directory and import those as external roles into Fortanix DSM. After importing the LDAP groups to Fortanix DSM, Group Administrators can map the external roles to that group by specifying a desired access level.

For example, if an Active Directory group identified by the distinguished name `CN=My Group,CN=Users,DC=example,DC=com` is added as an external role, an Administrator of group Example can map that external role to Example with access level Group Auditor. When a user that belongs to this Active Directory group authenticates to Fortanix DSM through LDAP, the user's session will have auditor access to the Example group.

**NOTE**:

- The user must be an Account Member since Account Administrators and Auditors have default access to all groups.
- The user need not be added to the Example group directly, but the user will have access to the Example group based on Active Directory group membership. If the user is removed out of that Active Directory group, that user will lose access to the Fortanix DSM group.

### 8.5.2 AUTHORIZATION SETTINGS

When enabling LDAP authorization, the Account Administrator can specify how long an authorization is valid for. When an authorization expires, Fortanix DSM will query the LDAP directory for the user's current group memberships and update the user's session accordingly.

It is also possible to specify a required role for all users and apps of the account authenticating through LDAP. The required roles supported are Account Administrator, Account Auditor, Account Member, Regular Apps, and Administrative Apps. If a user or app

is not a member of this directory group and does not have the corresponding LDAP role, Fortanix DSM will prevent that user from selecting the account.

When authorization is enabled for an LDAP integration, the following settings are required:

- `Base DN`
- `User Object Class`

The `Service Account` setting is also needed if the directory does not allow anonymous search. Note that these settings are also applicable when DN resolution is set to `Search by Mail`.

### 8.5.3 ADDITIONAL REQUIREMENTS

To use the LDAP authorization mechanism, the LDAP directory must support identifying objects with unique ids with one of the following attributes:

- `entryUUID` defined in RFC 4530, supported by Open LDAP and others.
- `objectGUID` used by Active Directory

When comparing an external role against a user's LDAP groups, Fortanix DSM uses the group's unique id instead of its distinguished name. Unique ids are more flexible compared to DNs since changing object attributes does not affect its unique id, but may change its DN. For example, the group name is usually included in the DN.

### 8.5.4 LDAP IDENTITY PROVIDER REQUIREMENTS

The identity provider must:

- Conform to LDAPv3 protocol specified in RFC 4511 and other related RFCs.
- Either support `ldaps` scheme or, if using `ldap` scheme, the server must support the `StartTLS` extended operation.

**Warning:** Administrator lock-out: If the SSO mechanism is misconfigured, you will not be able to log in to your account. When updating the SSO configuration, make sure to

check the box for "**Account Administrators can log in with password**" option. This way, Account Administrators can still log in with a password when the SSO provider is unavailable.

## 8.6 LDAP BASED AUTHORIZATION FOR APPLICATIONS

Apps with LDAP authentication methods can have either normal authorization or use LDAP authorization. Authorization for an application extends the App's authorization model to tie it to group membership in an LDAP compliant directory.

We define an authorization method for apps with two possibilities: Native, and External Directory. Existing apps will default to Native. For the app with External Directory authorization, the group membership and authorization of the app are determined dynamically based on its group membership in LDAP.

For the External Directory option, we define the following parameters:

- `directory_id`: which LDAP directory to search (we support multiple LDAP integrations in each account).
- `app_dn`: The distinguished name of the app to search for.

**NOTE**: Apps must also exist in the LDAP compliant directory so Fortanix DSM can use it for determining authorization.

Similar to the "External Role" described in the previous section, we have "External App Authorization" which will do the following:

- Map LDAP group to a Fortanix DSM group. To Map LDAP group:

1. Go to the **EXTERNAL ROLES** tab to import External Roles from the LDAP directory.
2. Click **SEARCH DIRECTORY** to search the External Roles in the LDAP directory.
3. Select the external roles and click **IMPORT EXTERNAL ROLES**.
4. To map the External Role to an App, under the column **Group for apps** click **MAP TO GROUPS**.

Once you map to a group, define the permissions for all Apps that are part of this LDAP group using the permissions icon.

When an App authenticates successfully, Fortanix DSM will query the corresponding LDAP directory using a service account to find its group memberships and then using the defined "External App Authorization" it determines the following entitlements for this App:

- Groups that it has access to.
- Operations that it can perform using security objects in applicable groups.

For example:

Create a group called "**CodeSigning**" in Fortanix DSM that has a key named "**key_SoftwareSigning**".

Now create two separate apps, say "**app1**" and "**app2**" with authentication method "External Directory" whose authorization is determined by External mapping.

**NOTE**: Name the app to have full distinguished name of the app in the Active Directory.

These apps must also already exist in the LDAP directory groups "**AMR\DevTeam21"** and "**AMR\KeyManagementTeam"** respectively.

Define "External App Authorizations" by searching for appropriate object class (for example Group) as follows:

Search for the group "**AMR\DevTeam21"** and map it to the Fortanix DSM group "**CodeSigning"**, and set only the "**Sign**" permission for the app such that the app can only perform the Sign operation and will disallow other operations on the Security Object.

Similarly, search for the group "**AMR\KeyManagementTeam"** and map it to the Fortanix DSM group "**CodeSigning"**, and set only the "**UnWrap**" permission for the app such that the app can only perform the UnWrap operation and will disallow other operations on the Security Object.

When the "**app1**" tries to authenticate to Fortanix DSM using the applicable authentication mechanism, it will dynamically determine authorization entitlements for this app by querying the LDAP directory. So, "**app1**" will show up with membership of group "**AMR\DevTeam21**" and will have the entitlement to use the Security Objects in Fortanix DSM group "**CodeSigning**" but will only be used to perform "**Sign**" operation using those objects.

This way LDAP directory continues to be the source of truth for authorization and by removing an application from a group in LDAP directory, the authorization can be easily revoked.

## 8.7 AUTHORIZATION FOR PLUGINS

Similar to the Fortanix DSM apps, the Fortanix DSM plugins are entities that may be in multiple groups.

A plugin may use a security object if and only if it shares a group with the security object.

An app, user, or another plugin may invoke a plugin if and only if it shares a group with the plugin.

In a typical configuration, a plugin will be in a privileged group A that contains the keys the plugin will use, as well as a less privileged group B that contains the apps that will invoke the plugin. Since the apps in the less privileged group B are not in the privileged group A, the apps may only access the keys by invoking the plugin, which may enforce access control policies, invariants, and so on.

## 9.0 ACCESS CONTROL FOR KEYS

A key (or Security Object) in Fortanix DSM can be managed by a user or an application. This includes actions such as generating a key, importing a key, disabling a key, changing permissions on a key, or deleting a key. A key, however, can be used for cryptographic operations only by an application. A key resides in a group in Fortanix DSM, and it can be accessed only by users and applications that also belong to that group.

A typical use case for Fortanix DSM in a large enterprise may involve providing access to keys in several applications which operate on the same encrypted data set but are owned by separate business units or product owners. Fortanix DSM makes it very easy for applications to either share keys or keep access to their set of keys isolated.

- **Key sharing between applications**: Sharing keys between applications is as easy as making the applications member of the same group. An application in Fortanix DSM may be a member of multiple groups and has a default group which is used by the APIs if no group is explicitly specified. The application has full access to all the keys in all the groups it belongs to. For example, Application A has full access to all the keys that Application B generates due to its membership in Group B.
- **Partitioning of key space between applications:** Fortanix DSM also allows applications to keep their key space separated by making them members of disjoint set of groups. For example, Application C does not have access to any of the keys available to Applications A and B as it does not share any group with those applications.

The group membership of applications is determined either by the Account Administrator (AA), or the Account Member (AM) if the AM has a Group Administrator role for the group, he is providing access to. It is imperative on the Fortanix DSM users to ensure how applications are provided access to keys in Fortanix DSM.

# 10.0 DOCUMENT INFORMATION

## 10.1 DOCUMENT LOCATION

The latest published version of this document is located at the URLs:

https://support.fortanix.com/hc/en-us/articles/360033005052-User-s-Guide-Authorization

https://support.fortanix.com/hc/en-us/articles/360033272171-User-s-Guide-Authentication

## 10.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com