Secret Management

Advanced Conversational Engagement

Exported on 06/06/2025

Table of Contents

We have used Azure Key Vault to provide secure storage to generic secrets. These secrets can be called upon using the secret identifier. These secrets include connection string for Databases, primary and secondary key for storage account, clientId and objectId for Managed Identities, Service principal passwords, etc. There are two ways to push these secrets to key vault. We have created a BICEP module, which is called upon in a resource deployment to push resource related secrets into the key vault.

Following is the reference template

```
@description('The name of the Key Vault.')
@minLength(3)
@maxLength(24)
param pKeyVaultName string

@description('The name of the secret.')
@minLength(1)
@maxLength(127)
param pSecretName string

@description('The value of the secret.')
@secure()
param pSecretValue string

@description('The value of the Source tag')
param pSource string

param pSecretExpiration int = dateTimeToEpoch(dateTimeAdd(utcNow(), 'P1Y'))

var enabled = true

resource keyvault 'Microsoft.KeyVault/vaults@2021-10-01' existing = {
  name: pKeyVaultName
}
resource keyVaultSecret 'Microsoft.KeyVault/vaults/secrets@2021-10-01' = {
  name: pSecretName
  parent: keyvault
  tags: {
    source: pSource
  }
  properties: {
    attributes: {
      enabled: enabled
      exp: pSecretExpiration
    }
    value: pSecretValue
  }
}
```

We have used this same module to push the secrets using push key vault secret pipeline: https://dev.azure.com/cbsp-abnamro/GRD0001014/_build?definitionId=54982

1 Azure policies AAB Key Vault

There are two AAB azure policies which are added onto the key vault for secrets.

1. *Secrets should have more than the specified number of days before expiration:* If a secret is too close to expiration, an organizational delay to rotate the secret may result in an outage. Secrets should be rotated at a specified number of days prior to expiration to provide sufficient time to react to a failure. As a best practice, organizations should have sufficient time to rotate a Secret before it expires. Hence, Secrets should not be created when the number of days before expiration is '7' or less. For that reason, this Azure Policy denies the creation of a Secret when the 'Days to expiration' setting is set to '7' or less.
2. *Secrets should have the specified maximum validity period:* This Policy Set bundles all Policy Definitions associated to the Basic, Standard and Foundation controls for the Azure Key Vault Service. As a best practice, Secrets within an Azure Key Vault should only be valid for a maximum number of 365 days. Therefore, this Azure Policy denies the creation of a Secret when its validity period is above 365 days.

With the above two policies in place we have to add expiry date to all the secrets present in the key vault which is taken care at the time of creation of new secret when we set the vault of

pSecretExpiration date of creation + one year.

2 Secret Rotation

Once the expiry date of the secret is near i.e. if a secret is going to expire in 7 days or less, your Key Vault becomes non-compliant. To resolve this non compliancy we have to rotate or update the secret based on its value.

Now as mentioned our key vault contains secrets whos value do not change (connection strings, objectId's) as well as secrets who's values are updated at a regulars period of time(SPN passwords).

2.1 Generic secret rotation

To manage these secrets we have added tags to them based on the source though which they were pushed. For example, secrets pushed at the time of resource creation i.e. connection strings, objectId's, clientId's will have the tag **source: InfraSecret.** Similarly secrets pushed using push key vault secret pipeline will have tag **source: ApplicationSecret.**

As mentioned, the values of the secrets having tags InfraSecret and ApplicationSecret do not change. So to manage those we update the expiration date of these secrets. This is done using a scheduler which runs everyday 5:00 AM CET. This scheduler runs a PowerShell script which filters all the secrets in the Key vault based on the tags and checks the expiry date to the date when the scheduler is running. If the secret does not have an expiry date or if the expiry date of the secret is less the 31 days, then we update the expiry date of the secret to current date + 365 days.

```
                $setExpiryDate = (Get-Date).AddYears(1)
                $secrets = Get-AzKeyVaultSecret -VaultName $
{{ variables.keyVaultName }}
                # Update Expiry on Secrets
                #Below code adds expiry to secret if expiry is empty and updates the
expiry if expiry date is less than or equal to 31 days
                foreach ($secret in $secrets)
                {
                  if(($secret.Tags.source -eq 'infrasecret') -or ($secret.Tags.source
-eq 'applicationsecret'))
                  {
                    $currentSecret = Get-AzKeyVaultSecret -VaultName $
{{ variables.keyVaultName }} -Name $secret.Name
                    $currentExpiry = $secret.Expires
                    if ($currentExpiry -le (Get-Date).AddDays(31) -or (-not
$currentExpiry))
                      {
                        Set-AzKeyVaultSecret -VaultName ${{ variables.keyVaultName }}
-Name $currentSecret.Name -SecretValue $currentSecret.SecretValue -Expires
$setExpiryDate -Tag $currentSecret.Tags
                        Write-Output "Secret: $($secret.Name) is less than 31 days
hence updating the expiry date as $setExpiryDate"
                      }
                    else
                      {
                        Write-Output "Secret: $($currentSecret.Name) already has a
sufficient expiry date: $currentExpiry"
                      }
                  }
                }
```

2.2 SPN secret Rotation

After the creation of service principal, the ApplicationId, ObjectId and password of the SPN is pushed into the key vault. This SPN password should be Renewed before expiry. To do this we have created an spn password rotation pipeline which first ckecks if the spn is expiring of not, if yes then by using the FSCP function app API it updates the password. This new password along with the spn appId and objectId is again pushed into the key vault. After the new password is updated in the Key vault, the old expired password is clean up from Microsoft Entra ID.

Confluence: Service Principal Password Rotation - Advanced Conversational Engagement - Confluence[1]

[1] https://confluence.int.abnamro.com/pages/viewpage.action?spaceKey=CIEDC&title=Service+Principal+Password+Rotation