Key Vault Associated to Wrong TenantID_Encryption

Last updated by | Kevin Gregoire | Mar 29, 2022 at 11:47 AM PDT

Tags

cw.Azure-Encryption	cw.TSG

Contents

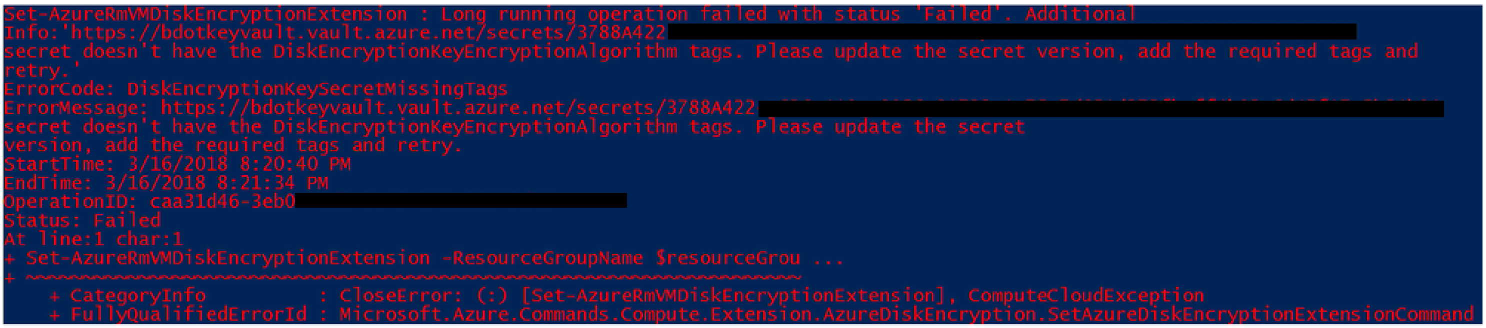
Scenario:

The customer attempts to encrypt their VM, but they get an error stating the following:

```
Set-AzureRmVMDiskEncryptionExtension : Long running operation failed with status 'Failed'. Additional  Info:'h
```

This error is generated because the secret is not pulling the correct information from the AAD. The KeyVault has the incorrect tenant ID associated to it. This is caused when a KeyVault is moved between subscriptions or the Tenant ID for the subscription has changed.

Screenshot showing the error:



```
Set-AzureRmVMDiskEncryptionExtension : Long running operation failed with status 'Failed'. Additional
Info:'https://bdotkeyvault.vault.azure.net/secrets/3788A422[redacted]
secret doesn't have the DiskEncryptionKeyEncryptionAlgorithm tags. Please update the secret version, add the required tags and
retry.'
ErrorCode: DiskEncryptionKeySecretMissingTags
ErrorMessage: https://bdotkeyvault.vault.azure.net/secrets/3788A422[redacted]
secret doesn't have the DiskEncryptionKeyEncryptionAlgorithm tags. Please update the secret
version, add the required tags and retry.
StartTime: 3/16/2018 8:20:40 PM
EndTime: 3/16/2018 8:21:34 PM
OperationID: caa31d46-3eb0[redacted]
Status: Failed
At line:1 char:1
+ Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $resourceGrou ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : CloseError: (:) [Set-AzureRmVMDiskEncryptionExtension], ComputeCloudException
    + FullyQualifiedErrorId : Microsoft.Azure.Commands.Compute.Extension.AzureDiskEncryption.SetAzureDiskEncryptionExtensionCommand
```

Mitigation Process:

When you create a new key vault in a subscription, it is automatically tied to the default Azure Active Directory tenant ID for that subscription. All access policy entries are also tied to this tenant ID. When you move your

Azure subscription from tenant A to tenant B, your existing key vaults are inaccessible by the principals (users and applications) in tenant B. To fix this issue, you need to:

1. Change the tenant ID associated with all existing key vaults in this subscription to tenant B.

2. Remove all existing access policy entries.

3. Add new access policy entries that are associated with tenant B.

1) For example, if you have key vault 'myvault' in a subscription that has been moved from tenant A to tenant B, here's how to change the tenant ID for this key vault and remove old access policies.

```
$Select-AzureRmSubscription -SubscriptionId YourSubscriptionID
$vaultResourceId = (Get-AzureRmKeyVault -VaultName myvault).ResourceId
$vault = Get-AzureRmResource –ResourceId $vaultResourceId -ExpandProperties
$vault.Properties.TenantId = (Get-AzureRmContext).Tenant.TenantId
$vault.Properties.AccessPolicies = @()
Set-AzureRmResource -ResourceId $vaultResourceId -Properties $vault.Properties
```

Because this vault was in tenant A before the move, the original value of **$vault.Properties.TenantId** is tenant A, while **(Get-AzureRmContext).Tenant.TenantId** is tenant B.

2) To remove the existing policies from the VM, run the following script to clear the old configuration. Save the following syntax into notepad and save as Set-EncryptionSettings.ps1. You can then run in an Azure PowerShell window.

```
Param(
 [Parameter(Mandatory = $true)]
 [ValidateNotNullOrEmpty()]
 [string]$ResourceGroupName,
 [Parameter(Mandatory = $true)]
 [ValidateNotNullOrEmpty()]
 [string]$VmName,
 [Parameter(Mandatory = $false)]
 [ValidateNotNullOrEmpty()]
 [string]$DiskEncryptionSecretUrl,
 [Parameter(Mandatory = $false)]
 [ValidateNotNullOrEmpty()]
 [string]$DiskEncryptionSecretVaultId,
 [Parameter(Mandatory = $false)]
 [ValidateNotNullOrEmpty()]
 [string]$KeyEncryptionKeyUrl,
 [Parameter(Mandatory = $false)]
 [ValidateNotNullOrEmpty()]
 [string]$KeyEncryptionKeyVaultId
)
$vm = Get-AzureRmVm -ResourceGroupName $resourceGroupName -Name $vmName;
$backupEncryptionSettings = $vm.StorageProfile.OsDisk.EncryptionSettings;
if($backupEncryptionSettings)
{
   Write-Host "Encryption Settings of the VM before update:"
   Write-Host "EncryptionEnabled: $($backupEncryptionSettings.Enabled)";
   if($backupEncryptionSettings.Enabled -eq $true)
   {
       Write-Host "DiskEncryptionSecretUrl: $($backupEncryptionSettings.DiskEncryptionKey.SecretUrl)";
       Write-Host "DiskEncryptionSecretVaultId: $($backupEncryptionSettings.DiskEncryptionKey.SourceVault.Id)"
   }
   if($backupEncryptionSettings.KeyEncryptionKey)
   {
       Write-Host "KeyEncryptionKeyUrl: $($backupEncryptionSettings.KeyEncryptionKey.KeyUrl)";
       Write-Host "KeyEncryptionKeyVaultId: $($backupEncryptionSettings.KeyEncryptionKey.SourceVault.Id)";
   }
}
Write-Host "Stopping VM before updating VM model"
Stop-AzureRmVM -Name $vmName -ResourceGroupName $resourceGroupName;
$vm = Get-AzureRmVm -ResourceGroupName $resourceGroupName -Name $vmName;
##Clear encryption settings from VM model
Write-Host "Clearing encryption settings from VM model"
$vm = Get-AzureRmVm -ResourceGroupName $resourceGroupName -Name $vmName;
$vm.StorageProfile.OsDisk.EncryptionSettings.Enabled = $false;
$vm.StorageProfile.OsDisk.EncryptionSettings.DiskEncryptionKey = $null;
$vm.StorageProfile.OsDisk.EncryptionSettings.KeyEncryptionKey = $null;
Update-AzureRmVM -VM $vm -ResourceGroupName $resourceGroupName;
##Set input encryption settings to VM model
Write-Host "Setting input encryption settings To VM model"
$vm = Get-AzureRmVm -ResourceGroupName $resourceGroupName -Name $vmName;
if($DiskEncryptionSecretUrl -and
  $DiskEncryptionSecretVaultId)
{
   $vm.StorageProfile.OsDisk.EncryptionSettings = New-Object Microsoft.Azure.Management.Compute.Models.DiskEnc
   $vm.StorageProfile.OsDisk.EncryptionSettings.enabled = $true;
   $vm.StorageProfile.OsDisk.EncryptionSettings.DiskEncryptionKey = New-Object Microsoft.Azure.Management.Comp
   $vm.StorageProfile.OsDisk.EncryptionSettings.DiskEncryptionKey.SecretUrl = $DiskEncryptionSecretUrl;
   $vm.StorageProfile.OsDisk.EncryptionSettings.DiskEncryptionKey.SourceVault = New-Object Microsoft.Azure.Man
   $vm.StorageProfile.OsDisk.EncryptionSettings.DiskEncryptionKey.SourceVault.Id = $DiskEncryptionSecretVaultI
   if($KeyEncryptionKeyUrl -and
      $KeyEncryptionKeyVaultId)
   {
       $vm.StorageProfile.OsDisk.EncryptionSettings.KeyEncryptionKey = New-Object Microsoft.Azure.Management.C
       $vm.StorageProfile.OsDisk.EncryptionSettings.KeyEncryptionKey.KeyUrl = $KeyEncryptionKeyUrl;
       $vm.StorageProfile.OsDisk.EncryptionSettings.KeyEncryptionKey.SourceVault = New-Object Microsoft.Azure.
       $vm.StorageProfile.OsDisk.EncryptionSettings.KeyEncryptionKey.SourceVault.Id = $KeyEncryptionKeyVaultId
   }
}
```

```
Update-AzureRmVM -VM $vm -ResourceGroupName $resourceGroupName;
$vm = Get-AzureRmVm -ResourceGroupName $resourceGroupName -Name $vmName;
##Start the VM
Write-Host "Starting VM with updated encryption settings"
$vm | Start-AzureRmVm;
```

3) Now that your vault is associated with the correct tenant ID and old access policy entries are removed, set new access policy entries with

```
Set-AzureRmKeyVaultAccessPolicy.
```

Need additional help or have feedback?

To engage the Azure Encryption SMEs...	*To provide feedback on this page...*	*To provide kudos on this page...*
Please reach out to the **Azure Encryption SMEs** for faster assistance. Make sure to use the **Ava process** for faster assistance.	Use the **Azure Encryption Feedback** form to submit detailed feedback on improvements or new content ideas for Azure Encryption. ***Please note*** the link to the page is required when submitting feedback on existing pages! If it is a new content idea, please put N/A in the Wiki Page Link.	Use the **Azure Encryption Kudos** form to submit kudos on the page. Kudos will help us improve our wiki content overall! ***Please note*** the link to the page is required when submitting kudos!