TLTurn on Transparent Data Encryption using your own key from Key Vault Using PowerShell

Last updated by | Soma Jagadeesh | Jan 10, 2021 at 9:46 PM PST

Contents

Turn on Transparent Data Encryption using your own key from Key Vault Using PowerShell

Prerequisites

1. You must have an Azure subscription and be an administrator on that subscription
2. [Recommended but Optional] Have a Hardware Security Module (HSM) or local key store for creating a local copy of the TDE Protector key
3. You must have Azure PowerShell version 4.2.0 or newer installed and running.
4. Create an Azure Key Vault and Key to use for TDE
   a. Azure Portal
   b. PowerShell instructions
      - Instructions for using a hardware security module (HSM) and Key Vault
5. The key must have the following attributes in order to be a TDE Protector:
   1. No expiration date
   2. Not disabled
   3. The key is able to do get, wrap key, and unwrap key operations.

1. Assign an AAD identity to your server

If you have an existing server, use the -Identity parameter to add an AAD identity to your server:

```
$server = Set-AzureRmSqlServer -ResourceGroupName <SQLDatabaseResourceGroupName>    -ServerName <LogicalServerName> -IdentityType SystemAssigned
```

If you are creating a new server, use the following with the -Identity parameter to add an AAD identity during server creation:

```
$server = New-AzureRmSqlServer -ResourceGroupName <SQLDatabaseResourceGroupName> -Location <RegionName> -ServerName <LogicalServerName> -ServerVersion "12.0" -SqlAdministratorCredentials <PSCredential> -IdentityType SystemAssigned
```

2. Grant Key Vault permissions to your server

Your server needs access to the key vault before using a key from it for TDE.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName <KeyVaultName> -ServicePrincipalName $server.Identity.PrincipalId -PermissionsToKeys get, wrapKey, unwrapKey
```

3. Add the Key Vault key to the server & Set the TDE Protector

Note: The combined length for the key vault name and key name cannot exceed 94 characters.

Note: An example KeyId from Key Vault is: https://contosokeyvault.vault.azure.net/keys/Key1/1a1a2b2b3c3c4d4d5e5e6f6f7g7g8h8h

Add the key from Key Vault to the server

```
Add-AzureRmSqlServerKeyVaultKey -ResourceGroupName <SQLDatabaseResourceGroupName> -ServerName <LogicalServerName> -KeyId <KeyVaultKeyId>
```

Set the key as the TDE protector for all resources under the server

```
Set-AzureRmSqlServerTransparentDataEncryptionProtector -ResourceGroupName <SQLDatabaseResourceGroupName> -ServerName <LogicalServerName> -Type AzureKeyVault -KeyId <KeyVaultKeyId>
```

To confirm that the TDE protector was configured as intended:

```
Get-AzureRmSqlServerTransparentDataEncryptionProtector -ServerName <LogicalServerName> -ResourceGroup <SQLDatabaseResourceGroupName>
```

4. Turn on TDE

```
Set-AzureRMSqlDatabaseTransparentDataEncryption -ServerName <LogicalServerName> -ResourceGroupName <SQLDatabaseResourceGroupName> -DatabaseName <DatabaseName> -State "Enabled"
```

Now the database or data warehouse has TDE enabled with an encryption key in Key Vault.

5. Check the encryption state and encryption activity of the database or data warehouse

Get the encryption state

```
Get-AzureRMSqlDatabaseTransparentDataEncryption -ServerName <LogicalServerName> -ResourceGroup <SQLDatabaseResourceGroupName> -DatabaseName <DatabaseName>
```

Check the encryption progress for a database or data warehouse

Get-AzureRMSqlDatabaseTransparentDataEncryptionActivity -ServerName <LogicalServerName> -ResourceGroup <SQLDatabaseResourceGroupName> -DatabaseName <DatabaseName>

Other useful PowerShell cmdlets

- Turning off TDE:

```
Set-AzureRMSqlDatabaseTransparentDataEncryption -ServerName
<LogicalServerName> -ResourceGroup <SQLDatabaseResourceGroupName> -
DatabaseName <DatabaseName> -State "Disabled"
```

- Returning the list of Key Vault keys added to the server:

```
<# KeyId is an optional parameter, to return a specific key version
#>

Get-AzureRmSqlServerKeyVaultKey -ServerName <LogicalServerName> -
ResourceGroup <SQLDatabaseResourceGroupName>
```

- Removing a Key Vault key from the server:

```
<# The key set as the TDE Protector cannot be removed. #>

Remove-AzureRmSqlServerKeyVaultKey -KeyId <KeyVaultKeyId> -
ServerName <LogicalServerName> -ResourceGroup
<SQLDatabaseResourceGroupName>
```

Troubleshooting

Check the following if an issue occurs:

- If the key vault cannot be found, make sure you’re in the right subscription.
  - Select-AzureRmSubscription -SubscriptionId <SubscriptionId>
- If the new key cannot be added to the server, or the new key cannot be updated as the TDE Protector, check the following:
  - The key should not have an expiration date
  - The key must have the get, wrap key, and unwrap key operations enabled

Classification

Root cause Tree - Security/User Request/How-to/advisory

How good have you found this content?