Azure Security and Compliance – Practical Exercises

Overview

This course includes optional practical exercises where you can try out the technologies described in the course for yourself. This guide lists the steps for the individual practical exercises.

See the Overview page under Practical Exercises in your course for information about getting started.

Setup

If you already have a Microsoft Azure subscription, you can skip this section. Otherwise, follow these steps to create a free trial subscription. You will need to provide a valid credit card number for verification, but you will not be charged for Azure services – for more information, see the frequently asked questions on the Azure sign-up page.

1. If you already have a Microsoft account that has not already been used to sign up for a free Azure trial subscription, you're ready to get started. If not, don't worry, just create a new Microsoft account.
2. After you've created a Microsoft account, create your free Microsoft Azure account. You'll need to sign-in with your Microsoft account if you're not already signed in. Then you'll need to:
    - Enter your cellphone number and have Microsoft send you a text message to verify your identity.
    - Enter the code you have been sent to verify it.
    - Provide valid payment details. This is required for verification purposes only – your credit card won't be charged for any services you use during the trial period, and the account is automatically deactivated at the end of the trial period unless you explicitly decide to keep it active.

Install Azure PowerShell (if needed)

In this exercise, you will install Azure PowerShell.

Note: If you have already installed Azure PowerShell you can skip this exercise.

1. From your computer, open an elevated PowerShell prompt.

Cmdlets for Resource Manager

2. Run the **Install-Module AzureRM** command. This will install the AzureRM module which represents resource management.
3. If you get prompted to install and import the NuGet provider, Type **Y** and then press the **Enter** key.
4. If you are notified that the repository is untrusted, confirm that you want to install the modules by typing **Y** and then pressing the **Enter** key. The installation process will take several minutes as packages are downloaded and installed.
5. After the download and installation is finished, run the **Import-Module AzureRM** command.
6. **Note**: If you receive a message about running scripts on your computer has been disabled, temporarily change the execution policy:
   Set-ExecutionPolicy Unrestricted
   After the import command is complete, return the execution policy to restricted.
   Set-ExecutionPolicy Restricted

Cmdlets for Service Manager (Classic) – also includes basic cmdlets such as subscription management

1. Run the **Install-Module Azure** command. This will install the Azure module which represents service management.
2. If you are notified that the repository is untrusted, confirm that you want to install the modules by typing **Y** and then pressing the **Enter** key.
3. Once the download and installation is finished, run the **Import-Module Azure** command.

Explore storage cmdlets and update the Help pages

1. Run **Get-Command *azurestorage*** to view storage cmdlets you can use in this course.
2. This is a good time to run **Update-Help** so you have the latest help pages. Don't be concerned if some libraries don't update. You can always find the Help pages on TechNet.

> If you have trouble installing the PowerShell modules from the PowerShell gallery, you can try the WebPI method instead. Visit http://aka.ms/webpi-azps to download and install the modules.

Module 2 – Key Vault

Create a Key Vault (Portal)

In this exercise, you will create a new key vault in the Azure portal.

1. Navigate to the Azure portal and sign in.
2. On the Hub menu, click **More Services**.
3. Type **key vault** in the filter to reveal the available option for managing cryptographic information in the Azure Portal. Mark Key vaults as a favorite to pin it to your Hub menu.
4. Click **Key vaults**. If you have any existing key vaults they will appear in this list.
5. On the Key vaults blade, click **Add**.
6. On the Create Key Vault blade, fill in the following values to create a new key vault. Click **Create** when you are finished entering the information.
   - Name: **KeyVault-<RandomAlphaNumericString>** where <RandomAlphaNumericString> is a random assortment of letters and numbers to make the name unique across Azure.
   - Subscription: **<YourSubscription>**
   - Resource Group: **Create a new resource group named KeyVaultRG**
   - Location: **<YourLocation>**
   - Pricing tier: **P1 Premium**
   - Access policies: **Leave as default**
   - Advanced access policy: **Enable all options**
7. On the menu bar, monitor the alerts for progress as the new key vault is created.
8. On the Hub menu, click **Key vaults**. Confirm that the new key vault has been created.
9. Select your key vault.
10. Take a few minutes to explore the various configuration options that are available. For example, **Access control (IAM)**, **Keys**, **Secrets**, **Access policies**, and **Advanced access policies**. These areas will be explored the following labs for this module.

Add Security to the Key Vault

In this exercise, you will create a new group and assign key and secret permissions.

1. Navigate to the Azure Portal and sign in.
2. On the Hub menu, click **Azure Active Directory**.
3. On the Azure Active Directory blade, click **Users and groups.**
4. On the Users and groups blade, under MANAGE, click **All groups**.
5. Click **+ Add** to create a new group with the following information:
   - Name: **Network Operations Team**
   - Description:**<YourDescription>**
   - Membership type: **Assigned**
   - Members:**<Add yourself to the group>**
6. Click **Create** to create the new Azure Active Directory group.
7. On the Dashboard, navigate to your key vault.
8. On the Key vault blade, under SETTINGS, click **Access policies**.
9. Click **Add new** to create a new Access policy with the following information:
   - Select principal: **Network Operations Team**
   - Configure from template (optional): <leave as default value>
   - Key permissions: **Get and List**
   - Secret permissions: **Get and List**
   - Authorized application: **None selected**
10. Click **OK** to add the key vault access policy.
11. On the Key vault blade, under SETTINGS, click **Access control (IAM)**.
12. Click **+ Add** and review the **Roles** that are available. Specifically, notice the **Reader** and **Key Vault Contributor** roles.
13. Select **Reader** and **Network Operation Team** as a user.
14. Confirm the **Network Operation Team** the **Key Vault Contributor** role.

Create a Key (Portal)

In this exercise, you will create a new key using the Azure portal and PowerShell.

Create a key in the portal

1. Navigate to the Azure portal and sign in.
2. Navigate to your key vault.
3. Under Assets, click **Keys**.
4. On the Keys blade, click **+ Add**.
5. Notice the three options to create a key: **Generate**, **Upload**, and **Restore Backup**.
6. On the Create a key blade, fill in the following values to create a new key. Click **Create** when you are finished entering the information.
   - Options: Generate
   - Name: **Key1**
   - Key Type: **Software key**
   - Set activation date: **Unchecked**
   - Set expiration date: **Unchecked**
   - Enabled: **Yes**
7. On the menu bar, monitor the alerts for progress as the new key is created.
8. Confirm the key was added to the key vault.

Create a key using PowerShell

1. Open an elevated PowerShell cmd prompt.
2. Login to Azure. In the Sign in to your account window, enter your Azure administrative credentials and then click **Sign in**. Complete your authentication as needed (for example, if you have two-factor authentication enabled, you might be prompted for the second authentication factor).
   Login-AzureRmAccount
3. Generate a software key and store it in a variable.
   $key2 = Add-AzureKeyVaultKey -Name Key2 -VaultName <your key vault> -Destination software
4. View the URI for the key.
   $key2.id
5. Return to the portal and confirm your new key was created. Notice the Key Identifier URI.
6. Notice that all the operations are permitted.

7. Return to PowerShell and disable the key.
   Set-AzureKeyVaultKeyAttributes -Name Key2 -VaultName <your key vault> -Enable $false
8. Verify in the portal that the key is disabled. You may need to **Refresh** the page.

Create a Secret (Portal and PowerShell)

In this exercise, you will create a new secret using the Azure portal and PowerShell.

Create a secret in the portal

1. Navigate to the Azure portal and sign in.
2. Locate your key vault.
3. Under Assets, click **Secrets**.
4. On the Secrets blade, click **+ Add**.
5. Notice the upload options: **Certificate** and **Manual**.
6. On the Create a secret blade, fill in the following values to create a new secret. Click **Create** when you are finished entering the information.
   - Options: **Manual**
   - Name: **DBConnectionString1**
   - Value: **Pa$$w0rd**
   - Content type: **Leave blank**
   - Set activation date: **Unchecked**
   - Set expiration date: **Unchecked**
   - Enabled: **Yes**
7. On the menu bar, monitor the alerts for progress as the new secret is created.
8. Confirm the secret was added to the key vault.

Create a secret using PowerShell

1. Open an elevated PowerShell cmd prompt.
2. Login to Azure. In the Sign in to your account window, enter your Azure administrative credentials and then click **Sign in**. Complete your authentication as needed (for example, if you have two-factor authentication enabled, you might be prompted for the second authentication factor).
   Login-AzureRmAccount
3. Create a secure string for the secret value. This is a connection string to the Northwind database.
   $SecretValue = ConvertTo-SecureString 'Data Source=.\;Initial Catalog=Northwind;Integrated Security=True' -AsPlainText –Force command.
4. Create the secret and store it in a variable.
   $Secret = Set-AzureKeyVaultSecret -VaultName '<YourKeyVault>' -Name 'DbConnectionString2' -SecretValue $SecretValue

5. You can now reference the secret that you added to Azure Key Vault by using its URI.
   $secret.Id
6. View the secret.
   Get-AzureKeyVaultSecret -VaultName <YourKeyVault>
7. Return to the portal and confirm your secret was created.

Key Vault Diagnostic Logging (Portal and PowerShell)

In this exercise, you will learn about Key Vault logging.

Explore diagnostic logging

1. Navigate to the Azure portal and sign in.
2. Locate your key vault.
3. Select **Diagnostic Logging** and notice the logging is **Off**.

Enable diagnostic logging

4. Open an elevated PowerShell cmd prompt.
5. Login to Azure. In the Sign in to your account window, enter your Azure administrative credentials and then click **Sign in**. Complete your authentication as needed (for example, if you have two-factor authentication enabled, you might be prompted for the second authentication factor).
   Login-AzureRmAccount
6. Create a storage account for the diagnostic logs. The name must be unique, and you should use the same datacenter location as your key vault. This may take a minute to complete.
   $sa = New-AzureRmStorageAccount -ResourceGroupName <your resource group> -Name keyvaultlogs -Type Standard_LRS -Location <YourLocation>
7. Retrieve the key vault name.
   $kv = Get-AzureRmKeyVault -VaultName <YourKeyVaultName>
8. Enable diagnostic logging.
   Set-AzureRmDiagnosticSetting -ResourceId $kv.ResourceId -StorageAccountId $sa.Id -Enabled $True -Categories AuditEvent
9. The output will deplay the details of the StorageAccountId and log retention settings.
10. Return to the portal and verify **Diagnostics Logs** is now **On**. Notice your storage account name.
11. Now that diagnostics have been enabled, you can do further exploration on your own. Begin by creating some secrets and keys.
12. View the diagnostic log using the Get-AzureStorageBlob cmdlet or by browsing to the insights-logs-auditevent container in the storage account (you'll need to browse all the way down to the JSON file). To view the JSON file download the file and open in Visual Studio or a text editor.