Certificates and Keys

Certificates and Keys

TLS certificates and keys are used by the Multicloud Defense Gateway in proxy scenarios. For ingress (reverse proxy) users access the application via Multicloud Defense Gateway and it presents the certificate configured for the service. For egress (forward proxy) cases, the external host's certificate is impersonated and signed by the certificate defined.

Certificate body is imported to the Multicloud Defense Controller. The private key can be provided in the following ways:

- Import the private key contents.
- Store in AWS secrets manager and provide the secret name.
- Store in AWS KMS and provide the cipher text contents.
- Store in GCP secrets manager and provide the secret name.
- Store in Azure keyvault and secret and provide the keyvault and secret name.

For testing purposes you can also generate a self-signed certificate on the Multicloud Defense Controller. This is similar to importing the private key contents from your local file system.

Note Certificates are **NOT** editable once created. If you need to replace the existing certificate, you will need to create a new certificate, edit the decryption profile to reference the new certificate, and then delete the old certificate.

When importing the certificate and private key, the Multicloud Defense Controller / UI can detect if there is a mismatch. However, when using any other import method where the private key is stored within the cloud service provider, the Multicloud Defense Controller / UI will not be able to detect if there is a mismatch. This is by design to ensure the private key remains private and within your cloud service provider. When the private key is needed by the Multicloud Defense Gateway, it is accessed and used, and if there is a mismatch, an error is generated.

Import Certificate

Step 1 Navigate to **Mange** > **Security Policies** > **Certificates**.
Step 2 Click **Create**.
Step 3 When prompted with the **Method**, choose **Import your Certificate and Private Key**.
Step 4 Copy the contents of the certificate file in the **Certificate Body**. This can include the certificate and the chain.
Step 5 Copy the contents of the private key in **Certificate Private Key**.
Step 6 (Optional) Import the chain into the **Certificate Chain** if your certificate and the chain are in different files.
Step 7 Click **Save**.

AWS - KMS

Step 1 Navigate to **Mange** > **Security Policies** > **Certificates**.
Step 2 Click **Create**.
Step 3 In the Method choose *Import AWS - KMS*.
Step 4 Select the Cloud Account and the region.
Step 5 Copy the contents of the Certificate file in the *Certificate Body*. This can include the certificate and the chain.
Step 6 Copy the AWK KMS encrypted cipher text in the *Private Key Cipher Text*. .
Step 7 Click **Save**.

AWS - Secrets Manager

Step 1 Navigate to **Mange** > **Security Policies** > **Certificates**.
Step 2 Click **Create**.
Step 3 In the Method choose *Import AWS - Secret*.
Step 4 Select the Cloud Account and the region.
Step 5 Copy the contents of the Certificate file in the *Certificate Body*. This can include the certificate and the chain.
Step 6 Provide the Secret Name where the private key is stored. The private key contents must be stored as *Other type of Secrets* > *Plain Text* in the AWS Secrets Manager.
Step 7 Click **Save**.

Azure Key Vault

Step 1 Navigate to **Mange** > **Security Policies** > **Certificates**.

Step 2 Click **Create**.
Step 3 In the Method choose *Import Azure - Key Vault Secret*.
Step 4 Select the Cloud Account and the region.
Step 5 Copy the contents of the Certificate file in the *Certificate Body*. This can include the certificate and the chain.
Step 6 Provide the Key Vault Name and the Secret Name where the private key is stored.
Step 7 Click **Save**.

GCP - Secret Manager

Step 1 Navigate to **Mange** > **Security Policies** > **Certificates**
Step 2 Click **Create**
Step 3 In the Method choose *Import GCP - Secret*
Step 4 Select the Cloud Account
Step 5 Provide the Secret Name (full path) and the Secret Version
Step 6 Copy the contents of the Certificate file in the *Certificate Body*. This can include the certificate and the chain
Step 7 Click **Save**.

Server Certificate Validation

When the gateway acts as a forward proxy, server certificate validation is automatically included in traffic processing. A designated server certificate validation **action** is not required in order to process traffic but it can improve the general security. By default, server certificate validation is not enabled and traffic going to servers that may have an invalid server certificate passes. Enable a server certificate validation action to prioritize rules for traffic that should not be allowed, or for specific traffic that should be trusted even regardless of its server certificate validations state.

Note This validation process is **only** applicable for forward proxy environments and when **decryption** is enabled.

We recommend enable server certificate validation actions primarily in the TLS decrpytion profile for general rule actions. FQDN service objects can be modified to enable validation actions if you need to override the TLS decryption selection. You can include and enable a server certificate validation in two methods:

- Server Certificate Validation in the TLS Decryption Profile
- Server Certificate Validation in the FQDN Service Object

Server Certificate Validation in the TLS Decryption Profile

When you select an action for server certificate validation within a TLS decryption profile, this action is used in all the rule sets that use this decryption profile. By default the validation action is configured to allow all

traffic regardless of whether the server certificate is valid or not, and Multicloud Defense does not generate an alert within the HTTPs logs.

Note If you enable the validation check to **Log**, locate the logs in **Investigate** > **Flow Analytics** > **HTTPS Logs**.

Use the following procedure to enable the server certificate validation in the TLS decrpytion profile:

Step 1 From the Multicloud Defense Controller, navigate to **Manage** > **Profiles** > **Decryption**.

Step 2 Select the TLS decryption profile you want add the server certificate validation to. If you do not have a profile ready, create one here. See Decryption Profile for more information.

Step 3 **Edit** the decryption profile.

Step 4 Under the **Profile Properties** section, expand the **Invalid Server Certificate Action** drop-down.

Step 5 Select one of the following options:

- **Deny Log** - This option automatically drops connections that do not provide a validated server certificate and logs the incident.
- **Deny No Log** - This option automatically drops connections that do not provide a validated server certificate and **does not** log the incident.
- **Allow Log** - This option allows connections that do not provide a validated server certificate to pass and logs the incident.
- **Allow No Log** - This option allows connections that do not provide a validated server certificate to pass and **does not** log the incident. This is the default action selection.

Step 6 Click **Save**.

What to do next

Ensure the TLS decryption profile is correctly associated with a forward proxy service object. See Forward Proxy Service Object (Egress / East-West) for more information.

Once the TLS decrpytion profile is included in a service object, confirm that the rule order within the policy is ordered in a way that supports how you want traffic processed.

Server Certificate Validation in the FQDN Service Object

Invalid server certificate validation within the FQDN service object is optional. If specified it will override the behavior designated in the TLS decryption profile. If you do not specify a selection here, no additional action or override action is taken. You can use the invalid server certificate validation within the FQDN service object to block or allow traffic for a specific server that may otherwise be blocked or allowed by the TLS decryption profile.

Note that when you enable the validation check to **Log**, these logs are located in **Investigate** > **Flow Analytics** > **HTTPS Logs**.

Use the following procedure to include a server certificate validation action in a FQDN service object:

Step 1 From the Multicloud Defense Controller, navigate to **Manage** > **Security Profile** > **FQDNs**.

Step 2 Select the FQDN service object you want to modify.

Step 3 **Edit** the selected FQDN service object.

Step 4 In the list of FQDN service objects included in the ruleset, expand the **Invalid Server Certificate Action** drop-down menu and select one of the following options:

- **Deny Log** - Automatically drop connections that do not provide a validated server certificate and logs the incident.
- **Deny No Log** - Automatically drop connections that do not provide a validated server certificate and **does not** log the incident.
- **Allow Log** - Allow connections that do not provide a validated server certificate to pass and logs the incident.
- **Allow No Log** - Allow connections that do not provide a validated server certificate to pass and **does not** log the incident.

Step 5 Click **Save**.

What to do next

Ensure the FQDN service object is correctly associated with a rule or rule set. See Rule Sets and Rule Set Groups for more information.

Once the FQDN service object is successfully assocaited with a rule or rule set in your policy, confirm that the rule order within the policy is ordered in a way that supports how you want traffic processed.