Telemetry Security Configuration

Security for the Telemetry File Target

Remote locations in the Target Folder input for the File Target type must be within the corporate network and behind a firewall.

To prevent data loss, log files that are generated within the target directory for the File target are automatically archived daily in an Archives subfolder. Archiving files prevents log files and archived content from growing exponentially. The Archives folder has a limit of ten archived files, which automatically purges old files to prevent using too much system disk space.

Encryption for Telemetry Database Target

To safeguard the data from unauthorized copying of database files, the owner of the database used for storing job telemetry events must enable Data Encryption.

To prevent unrestricted growth of the target database, set up an automated process for purging old logs from the database. A Telemetry table is created in the telemetry target database. The Time column can be used to filter logs based on the time period.

Enable Data Encryption

Back up the certificate, key, and log database to ensure that you can restore them in the future.

1. **Create a certificate**
   You can use an existing certificate, a certificate that provisioned the SQL server, or create a new certificate. For more information about creating a certificate, see "Create Certificate" in Microsoft SQL Docs.

   Run the following script to create a master key and provision a certificate within the SQL server. The private key of the certificate will be encrypted with the master key.

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<PasswordHere>';
go
CREATE CERTIFICATE TelemetryDatabaseCert WITH SUBJECT = Telemetry;
```

2. **Create an encryption key**
   Run the following script to create an encryption key from the certificate for the telemetry database.

```
USE <TelemetryDatabaseName>;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_12321
ENCRYPTION BY SERVER CERTIFICATE TelemetryDatabaseCert;
GO
```

3. **Enable encryption**

   Run the following script to enable encryption.

```
USE <TelemetryDatabaseName>;
ALTER DATABASE <TelemetryDatabaseName>
SET ENCRYPTION ON;
```

4. **Back up the certificate and key**

   Run the following script to back up the certificate.

```
USE master;
GO
BACKUP CERTIFICATE TelemetryDatabaseCert
TO FILE = N'<Path>\TelemetryDatabaseCert.cer'
WITH PRIVATE KEY (
FILE = N'<Path>\TelemetryDatabaseCert_key.pvk',
ENCRYPTION BY PASSWORD = '<StrongPassword>'
);
GO
```

5. **Back up the database**

   Run the following script to back up the database.

```
USE <Telemetry DatabaseName>;
BACKUP DATABASE <Telemetry DatabaseName>
TO DISK = '<Path for database backup file>'
```

6. **(Optional) Restore the certificate**

   To restore the certificate, create a certificate from the backup files. Run the following script to restore the certificate.

```
USE master;
GO
CREATE CERTIFICATE TelemetryDatabaseCert
FROM FILE = '<Path>\TelemetryDatabaseCert.cer'
WITH PRIVATE KEY (
FILE = N'<Path>\TelemetryDatabaseCert_key.pvk',
```

```
DECRYPTION BY PASSWORD = '<StrongPassword>'
);
GO
```

7. **(Optional) Restore the database.**
   Run the following script to restore the database from the backup files.

```
RESTORE DATABASE <TelemetryDatabaseName>
FROM DISK = '<Path for database backup file>'
```