Module signing in SQL Server

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:33 AM PST

Contents

What is it

Like mentioned on ownership chaining, some commands will fail to execute, like TRUNCATE TABLE and Dynamic SQL.

One possible workaround is the usage of EXECUTE AS, in other words changing the execution context. However this as some security concerns since that in the event where the audit records are checked, what will appear is the impersonated user and not the caller of the command.

Another way of working around is with module signing.

How to implement

1. Create a certificate

```
USE MyDB
Go
CREATE CERTIFICATE MyCertificate
   ENCRYPTION BY PASSWORD = '<mypassword>'
   WITH SUBJECT = '<Certificate Subject>',
   EXPIRY_DATE = '01/01/2035';
GO
```

2. Create a certificate account

```
USE MyDB
GO
CREATE USER CertificateUser
   FROM CERTIFICATE MyCertificate
GO
```

3. Sign the stored procedure

```
USE MyDB
GO
ADD SIGNATURE TO MyStoredProcedure
   BY CERTIFICATE MyCertificate
    WITH PASSWORD = '<mypassword>'
GO
```

4. Grant the necessary permissions to certificate account (example, this user will need to execute the stored procedure and SELECT from a table)

```
USE MyDB
GO
GRANT SELECT
   ON dbo.tb1
   TO CertificateUser;
GO

GRANT EXECUTE
   ON MyStoredProcedure
   TO CertificateUser;
GO
```

5. Grant execute permission to the application user

```
GRANT EXECUTE
   ON MyStoredProcedure
   TO ApplicationUser
GO
```

Public Doc Reference

[Signing Stored Procedures with Certificate](#)

How good have you found this content?