Securing SQL Server: Securing CLR

CLR (Common Language Runtime) is a feature of SQL Server that allows you to create and run managed code within the database engine. Enabling CLR on SQL Server allows you to create stored procedures, triggers, and functions using languages such as C# and VB.NET.

The "CLR strict security" option is an important security feature of CLR that allows you to specify which CLR assemblies are allowed to be loaded into the database engine. When this option is enabled, only assemblies that are signed with a certificate or an asymmetric key and that are marked as SAFE or EXTERNAL_ACCESS can be loaded into the database.

Enabling CLR strict security can help to prevent malicious assemblies from being loaded into the database. This is because it requires assemblies to be signed with a certificate or key, which helps to ensure their authenticity and integrity. In addition, the SAFE and EXTERNAL_ACCESS options allow you to specify the level of access that an assembly has to external resources, which can help to prevent unintended or malicious actions.

To enable CLR strict security, you can use the following T-SQL command:

```
sp_configure 'clr strict security', 1
RECONFIGURE
```

This command will enable CLR strict security and will prevent unsigned or improperly marked assemblies from being loaded into the database.

the "CLR strict security" option is an important security feature of CLR that allows you to specify which CLR assemblies are allowed to be loaded into the database engine. By enabling this option, you can help to prevent malicious assemblies from being loaded into the database and protect your SQL Server installation from potential security risks.

Want to learn more about securing your SQL Server, take a look at my free white paper on sever steps to help harden your SQL Server.

Securing SQL Server Whitepaper.