剖析SQLServer2005SQLCLR代码安全性 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/259/2021_2022__E5_89_96_E6_9E_90SQLS_c98_259370.htm 提要 在SQL Server 2005内运行.NET框架代码是一件令人激动的事情还是一种威胁？本系列文章将全面探讨这类SQLCLR代码的安全问题，以便开发人员和DBA都能够有所借鉴。 一、 引言 编写运行于宿主在任何环境下的CLR中的.NET代码的主要优点之一是代码存取安全（CAS）。 CAS提供了一种基于代码的而不是基于用户的认证模式以预防各种代码的入侵问题。但是，这种安全模式如何与SQL Server 2005自己的新的增强的安全特征共存呢？默认情况下，你的.NET代码是比较安全的；但是，这两种安全模式很容易发生冲突而且容易给你带来一些问题。在本篇中，我们将简短地分析CAS幕后有关概念和在SQL Server 2005中新引入的一些安全特征；然后，在后面的几篇中分析如何实现在利用SQL Server所提供的高级可编程特征的同时，使这两种系统协同工作。 好消息是，为了实现SQL Server所提供的安全系统和通用语言运行时刻库协同工作，微软已经提供了一定的工具来实现代码控制。但是，也存在许多有趣的问题！ 能够用C#，VB或任何其它.NET语言编写存储过程及其它代码模块一直被长期期待，而这正是SQL Server 2005最激动人心的特征之一。开发人员和DBA最终都能够冲破存在于扩展存储过程的Transact-SQL（T-SQL）和C 中的羁绊，而用一种真正的具有高度生产力的语言编写数据库代码！ 同时，在数据库服务器的内存空间中运行.NET代码的前景吓坏了某些人，尤其是一些负责保护数据完整性并且确保这些服务器昼夜运行

的DBA们。运行一些开发人员的代码（能够完全存取.NET框架和Win32 API）的想法导致许多DBA坚持认为，对运行于服务器中的这样的代码进行维护根本超出他们的能力之外。 通过在会议上进行演讲并进行大量培训活动，以及我向同学和客户提问"是否在服务器上的.NET代码吓坏了他们及其原因"。最终得到下面一些典型的备受关注的问题： 含糊的安全问题。其中大多数与当前正在出现的攻击问题相关；但是，显然，对有哪些新内容还不理解更为关注。 需要学习一种全新的技能来评定是否代码是安全的。 在数据和代码之间存在很多的模糊性，特别是对于使用.NET代码创建用户定义的类型这种新的能力。 还有另一种方式能够实现代码与服务器的"混合"，尽管OLE自动化（SP_OS*）和命令外壳系统(xp_cmdshell)存储过程一直可用来让人们运行外部代码。事实上，在SQL Server 2005中的.NET框架代码，经常被称作是SQLCLR代码，因为它是基于.NET通用语言运行时刻库（CLR）。其实，它仅仅是另一种存在和运行于SQL Server内部的代码模块而已。它是新东西，而且很酷，但是也仍只是代码；但决不是T-SQL（仍然是首选的数据存取编码实现方式）的插件代替品；而是，SQLCLR代码为复杂的数据库应用程序开创了全新的可能性。迟早大多数的DBA都会使用它并且将不得不做出最后的决定-是否让它驻于数据库中。 在本文中，我将探讨人们对于SQLCLR代码最关心的一个问题之一：其安全性如何？实际上，我将故意模糊两种重要概念-安全性和可靠性。安全性意味着保持数据的安全，而可靠性意味着保持SQL Server的安全；可靠性经常被与安全性相混淆。因此，尽管我主要讨论安全问题，但是我还要涉及到一定的可靠

性问题。 我将假定，你熟悉在SQL Server 2005编写.NET代码的优点和基本知识。概括来说，包括下面这些内容： 程序集，作为打包、发布和版本管理的单元 .NET代码存取安全基础 SQL Server 2005的新的安全特征 换句话说，本文并不是一篇有关于SQLCLR代码的入门性文章。 二、 安全宿主SQLCLR代码 随着SQLCLR代码的引入，SQL Server 2005现在支持两种完全不同的运行时刻环境：好的旧的可靠的T-SQL和新的正在发展中的SQLCLR。在过去的几年中，T-SQL随着SQL Server版本的不断升级而不断发展，并且与存储在一个数据库中的数据和对象紧密集成到一起，也与SQL Server中的安全系统良好地集成。相比之下，SQLCLR代码，在内部使用了一种由CLR所提供的完全不同的安全系统，这是一种"温暖的"、安全的环境。在此环境下，代码的运行不是基于运行它的用户的安全资格而是基于代码本身的安全资格。同时，SQLCLR代码必须在数据库和服务器的安全范围内执行；然而，这两种安全系统是根本不同的。如今，微软的SQL Sever开发小组已经研究出一种方法使得这二者共存并能够协同工作。 能够在另外一个应用程序中可靠地安全地宿主CLR是.NET框架2.0的一种新特征。这种宿主环境及其SQL Server实现，正是使得这两种安全系统和平共处的"秘密"所在，因为宿主（在此是SQL Server）能够很大程度地控制运行的代码。这意味着，从一种安全角度来看，托管SQLCLR代码不被允许存取没有授权给它的数据库对象。该代码必须运行于用户会话的SQL Server安全上下文中，而且需要使用相关的与T-SQL代码相同的许可权来激活它。 注意 底线是，在一个数据库中，SQLCLR代码不能做比在相同的安全上下文中等价的T-SQL代码模块更多的

事情。 当设计怎样宿主CLR时，微软具有三个主要的设计目标： CLR及运行于其中的代码不能妥协于SQL Server的安全性和稳定性。 SQLCLR代码必须遵循SQL Server认证和授权规则。这在一定程度上意味着，它要运行于用户会话的安全上下文中。 系统管理员必须能够控制对操作系统资源的存取。这意味着，必须存在一种安全的方式来从SQL Server进程中存取机器资源。 这些目标的最明显的表现之一是，默认情况下，CLR集成是关闭的。如果你想在一个数据库中运行.NET代码，那么一个系统管理员必须把它打开。打开它的T-SQL代码需要使用sp_configure：sp_configure clr enabled，1GORECONFIGUREGO 当然，你还可以使用新的与SQL Server 2005一起安装的Surface Area配置工具来实现这一点，如图 1 所示。从Windows开始菜单下，选择"Microsoft SQL Server 2005→Configuration Tools→SQL Server Surface Area Configuration"，再选择"Surface Area Configuration for Features"，然后从列表下选择"CLR Integration"。 因此，正确理解开关CLR集成特征的含义是十分重要的；然而，它唯一影响的是，是否允许在存储过程、触发器、用户定义类型及用户定义函数中运行SQLCLR代码。如果它被禁用，那么，在该服务器实例中不会执行SQLCLR代码；如果它被启动，那么，任何CLR代码都可以执行（当然，假定用户拥有正确的执行权限）。如果它被禁用，它不会阻止你把SQLCLR程序集安装到数据库中。你可以安装所有你想使用的程序集（当然，假定你拥有这样做的属性许可权），但是它们在任何环境下都不会运行，直到你支持CLR集成为止。 当SQLCLR代码执行时，它是在一个严格的安全环境中-这是一个既能保护操作系统

资源又能保护SQL Server中的数据和对象的层。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com