计算机二级辅导:增强MIDAS的安全性 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/646/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_646948.htm 增强MIDAS的安全性 大家都知道，使用RemoteDataModule最令人头疼的就是安全性问题。 主要体现在： 1、远端只要知道应用服务器的端口号即可访问到应用服务器，而一旦访问到应用服务器，TClientDataSet即可获得ProviderNames列表。（观点：不让他轻易得到ProviderNames列表。） 2、一旦知道了ProviderNames列表，这就相当于将数据库暴露在外了。 例如：客户端可以通过SQL语句来对数据库进行操作了。（观点：我们的应用服务器根部不接受SQL语句。） 因为看到大家此类贴多如牛毛，又没有什么更好的解决方法，因此我发表一下我的拙见。 我对IAppServer接口进行了进一步的扩展，增强了RemoteDataModule的安全性。主要体现在： 1、客户端侦测到应用服务器的端口号可以建立与应用服务器的连接，但必须提供由TClientDataSet提供一GUID作为密码方能在设计阶段获得ProviderNames列表。此功能使得系统外部人员无法在设计阶段直接在TClientDataSet的ProviderName中直接获得应用服务器的TProvider实例。如果想通过IAppServer来获取ProviderNames列表则必须提供这一特定的GUID作为密码。 IAppServer的AS_GetProviderNames原形为 function AS_GetProviderNames: OleVariant. safecall. 扩展后的函数为 function AS_GetProviderNames(Password:WideString): OleVariant. safecall. 系统外部人员能够访问TRemoteDataModule的Provider的唯一方法就是猜测（或者成为蒙）出可能有

的ProviderName直接赋值给TClientDataSet的ProviderName属性。当然这是十分困难的（只要你不是直接将datasetProvider1作为TdatasetProvider的名称）。 2、虽然恶意者可能通过其他方法（包括猜测、穷举）来获取到一个具有较高权限的TProvider，但是此步的安全特性完全将其挡在了门外。TClientDataSet必须提供加密后的CommandText串才能得到应用服务的正确相应。因为这里的加密对象是SQL语句（一个字符串），所以可以使用n多种加密方法。如果应用服务器解密出的串为非法SQL串，会向客户端返回SQL语法错误信息。而我在处理时并没有对SQL进行真正的加密，而是在TClientDataSet的CommandText中包含了一特定的字符串作为钥匙，而如果服务器得到请求后在CommandText中没有找到这一钥匙则返回“ Missiong SQL property ”异常。如果服务器得到了这一钥匙，则将这一钥匙从CommandText串中移除后交给TProvider进行处理。 实现： 听上去好像很玄，但实现起来比较简单： 我这里简单说说对SQL串的加密方法： 打开Provider单元，找到TDataSetProvider的SetCommandText方法。 你应该明白了吧。。。 如果你比我还菜，你就这样写： var commandt:string. begin if CommandText=\ ’ \ ’ then Exit. if Copy(Commandtext,1,8) 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