MySQL数据库应该如何对抗解密高手 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/213/2021_2022_MySQL_E6_95_B0_E6_8D_c103_213945.htm 当你连接一个MySQL服务器时，你通常应该使用一个口令。口令不以明文在连接上传输。所有其它信息作为能被任何人读懂的文本被传输。如果你担心这个，你可使用压缩协议(MySQL3.22和以上版本)使事情变得更难。甚至为了使一切更安全，你应该安装ssh。用它，你能在一个MySQL服务器与一个MySQL客户之间得到一个加密的tcp/ip连接。 为了使一个MySQL系统安全，强烈要求你考虑下列建议： 对所有MySQL用户使用口令。记住，如果other_user没有口令，任何人能简单地用Mysql -u other_user db_name作为任何其它的人登录。对客户机/服务器应用程序，客户可以指定任何用户名是常见的做法。在你运行它以前，你可以通过编辑MySQL_install_db脚本改变所有用户的口令，或仅仅Mysql root的口令，象这样： shell> mysql -u root mysql mysql> 0update user set password=password("new_password") where user="root". mysql> flush privileges. 不要作为Unix的root用户运行MySQL守护进程。mysqld能以任何用户运行，你也可以创造一个新的Unix用户MySQL使一切更安全。如果你作为其它Unix用户运行mysqld，你不需要改变在user表中的root用户名，因为Mysql用户名与unix 用户名没关系。你可以作为其它unix用户编辑mysql.server启动脚本mysqld。通常这用su命令完成。 如果你把一个Unix root用户口令放在mysql.server脚本中，确保这个脚本只能对root是可读的。 检查那个运行Mysqld的Unix用

户是唯一的在数据库目录下有读/写权限的用户。 不要把process权限给所有用户。mysqladmin processlist的输出显示出当前执行的查询正文，如果另外的用户发出一个0update user set password=password("not_secure")查询，被允许执行那个命令的任何用户可能看得到。mysqld为有process权限的用户保留一个额外的连接, 以便一个mysql root用户能登录并检查，即使所有的正常连接在使用。不要把file权限给所有的用户。 有这权限的任何用户能在拥有mysqld守护进程权限的文件系统那里写一个文件!为了使这更安全一些，用0select ... into outfile生成的所有文件对每个人是可读的，并且你不能覆盖已经存在的文件。 file权限也可以被用来读取任何作为运行服务器的unix用户可存取的文件。这可能被滥用，例如，通过使用load data装载 “ /etc/passwd ” 进一个数据库表，然后它能用0select被读入。 如果你不信任你的dns，你应该在授权表中使用IP数字而不是主机名。原则上讲，--secure选项对mysqld应该使主机名更安全。在任何情况下，你应该非常小心地使用包含通配符的主机名。 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