安全管理方案

# 数据库系统安全相关因素

* 网络安全
* 操作系统安全
* 应用程序安全
* 数据库安全

# 1.网络安全

风险预防任务：

* 投资设置防火墙。
* 确保仅供授权的客户机进行访问。
* 限制服务器所使用的网络接口。
* 使用安全安装脚本：

mysql\_secure\_installation

* 遵守常规特权安全事项。
* 不通过Internet传输明文（未加密）数据。

MySQL 服务器在客户机/服务器环境中运行，并提供内在面向网络的服务。确保只有授权的客户机可以连接到服务器以访问其数据库是非常重要的。确保使用口令保护MySQL帐户并且不为其设置不必要的特权也非常重要。此外，请考虑限制服务器所使用的网络接口。有许多不错的免费开源防火墙。MySQL不是为了面向Internet。如果MySQL未放置在DMZ 中，应格外小心。事实上，如果MySQL与面向Internet的应用程序在同一计算机上运行，则可能应该仅使用文件套接字。

未加密的数据可被任何有时间和能力拦截该数据并使用它来达到自己目的的用户访问。因此，您应该使用SSL或SSH等加密协议。

# 2.操作系统安全

* 与配置的复杂程度有关：

将服务器主机上与MySQL不相关的任务数减至最少。其他的使用会导致潜在风险。

* 登录帐户对于专用于MySQL的计算机不是必需的。
* 完全专用于MySQL的系统可以在性能方面获益。

将不直接与运行MySQL相关的服务器主机任务数减至最少。当为主机配置的任务较少时，与运行复杂配置以支持多个服务的主机相比，前一个主机更易于确保安全。

所分配的MySQL 服务器最好是主要用于MySQL 或专供其用，而不是用于Web 托管或邮件处理之类的其他目的，或者是为通用交互使用而托管登录帐户的计算机。

如果其他用户可以登录，则可能会存在MySQL 安装及其管理帐户的私密数据库信息被公开的风险。例如，不正确的文件系统特权可能会导致数据文件被公开。用户可以运行ps命令来查看有关进程及其执行环境的信息。

当计算机仅用于MySQL 时，除了系统管理帐户和管理MySQL 自身可能需要的其他任何帐户（例如mysql 用户的帐户），无需再设置登录帐户。此外，如果在服务器主机上运行的网络服务较少，则需要保持打开状态的网络端口也较少。关闭端口可将主机可能遭遇的攻击途径减至最少。

尽量减少非MySQL 服务的数量意味着更多的系统资源可以专用于MySQL，从而可以在

性能方面获益。

**系统安全设置**：

* 服务器登录默认端口为22，需要修改为非默认端口
* 每三个月修改一次服务器密码
* 制定iptables策略

MYSQL服务器：

*-A INPUT -p tcp -m tcp --dport 数据库端口 -j ACCEPT*

*-A OUTPUT -p tcp -m tcp --sport 数据库端口 -j ACCEPT*

*-A INPUT -p tcp -m tcp --sport altas应用端口 -j ACCEPT*

*-A OUTPUT -p tcp -m tcp –dport altas应用端口 -j ACCEPT*

Atlas服务器：

*-A INPUT -p tcp -m tcp --dport altas应用端口 -j ACCEPT*

*-A OUTPUT -p tcp -m tcp --sport altas应用端口 -j ACCEPT*

*-A INPUT -p tcp -m tcp --sport web端口 -j ACCEPT*

*-A OUTPUT -p tcp -m tcp –dport web端口 -j ACCEPT*

* 扫描主机端口，检查除了mysql需要监听的端口之外，还有哪些端口是打开监听的，并去除不必要的端口
* 用非root用户运行mysql。

# 3.应用程序安全

用户可通过多种方法来损坏数据。您必须采取措施以保护数据免受如SQL注入等攻击。

* 请勿信任应用程序用户输入的任何数据。
* 用户可以使用具有特殊意义的字符（如引号或转义序列）获取应用程序代码。
* 如果用户输入类似于”;DROP DATABASE mysql;” 的内容，请确保应用程序保持安全。
* 保护数值和字符串数据的值。
* 否则，用户可以获取对安全数据的访问权限，然后提交可以销毁数据或导致服务器负载过高的查询。
* 甚至需要保护公开可用的数据。
* 攻击会浪费服务器资源。
* 保护Web 表单、URL名称、特殊字符等。

如果用户输入值234 时，应用程序生成一个类似于SELECT \* FROM table WHERE ID=234的查询，则用户可以输入值234 OR 1=1 使应用程序生成查询SELECT \* FROM table WHERE ID=234 OR 1=1。因此，服务器将检索表中的每一行。这将公开每一行并导致服务器负载过高。为了防止这种类型的攻击，请使用不会将值解释为SQL表达式的存储例程或预处理语句。

您可能认为：如果数据库仅包含公开可用的数据，则该数据库不需要保护。这是错误的。即使允许显示数据库中的任何行，仍应该通过注入INSERT、UPDATE、REPLACE 或DELETE 语句来防止拒绝服务攻击或修改数据的尝试。否则，数据将对合法用户不可用。

**以下是一些用于检测此问题的技巧**：

* 在所有Web 表单中输入单引号或双引号（“'”和“"”）。
* 修改动态URL，向其添加%22（“"”）、%23（“#”）和%27（“'”）。
* 在数值字段中输入字符、空格和特殊符号而不是数字。应用程序应删除这些内容再将其传递给MySQL；否则，将生成错误。

# 4.数据库安全

## 4.1 MySQL安装安全风险

* 网络安全：有关客户机帐户的信息存储在mysql数据库中。为每个帐户设置的特权应该仅提供对该帐户需要查看或修改的数据的访问权限。指定帐户口令可使用户难以使用别人的帐户连接到服务器。例如，MySQL root帐户具有可执行任何数据库操作的完全特权，因此有必要为其指定一个不易猜到的口令。
* 操作系统安全：通常使用专用的登录帐户管理MySQL。但是，该帐户所在的主机上可能还有其他登录帐户。将与MySQL不相关的帐户数量减至最少可以尽可能地降低此风险。
* 文件系统安全：目录和文件是文件系统的一部分，因此您需要保护它们免受在服务器主机上具有登录帐户的其他用户的直接访问。MySQL安装还包括用于管理和访问数据库的程序和脚本。用户需要能够运行其中的一部分（例如客户机程序），但应该无法对其进行修改或替换。

可以认为网络安全是最高防御级别。其中包含了操作系统安全。后者又包含了文件系统安

全以及更深层的用户安全。尽管每个安全级别都有缺陷（或漏洞），但是当它们结合使用

时，各个级别形成了一个几乎坚不可摧的堡垒。

* 默认情况下，MySQL 使用未加密的客户机/服务器连接。
* 尽快保持MySQL的标准配置。
* 未加密连接无法满足在网络上安全传输数据的要求。
* 网络通信容易受到监视和攻击。
* 在客户机和服务器之间传输的数据可能会被更改。
* 使用加密算法可抵御大多数威胁。
* 这些算法会使所有类型的数据无法读取。
* 这可抵御多种类型的攻击。
* 某些应用程序需要加密连接所提供的安全性，对于这些连接，必须进行额外的计算。

如果在客户机和服务器之间使用未加密的连接，则对网络拥有访问权限的用户将可以看到所有通信，并可以查看（和修改）正在发送或接收的数据。目前的行业惯例需要加密算法中的很多额外安全要素。这可抵御多种已知攻击，如更改加密消息的顺序或重放两次数据。数据加密是CPU密集型操作，需要计算机执行额外工作，因此可能推迟其他MySQL 任务。如果应用程序需要加密连接所提供的安全性，则必须进行额外的计算。

## 4.2 SSL

### 4.2.1 SSL 协议

MySQL 支持MySQL 客户机和服务器之间的SSL（secure

sockets layer，安全套接字层） 连接。

* SSL 连接协议具有以下特点：
* 使用不同的加密算法确保公共网络上数据的安全
* 检测任何数据更改、丢失或重放
* 结合了使用X509 标准提供身份验证的算法
* 需要额外安全保护（加密）的应用程序应使用SSL。
* 安全连接基于OpenSSL API。

MySQL 可以针对单个连接启用加密。根据各个应用程序的需要，您可以选择常规的未加密连接或安全的加密SSL 连接。

安全连接基于OpenSSL API，可通过MySQL C API 获取。复制时使用C API，因此可以在主服务器和从属服务器之间使用安全连接。

X509 使得在Internet 上进行身份识别成为可能。这主要用于电子商务应用程序中。基本上，应该有一个受信任的证书颁发机构(Certificate Authority, CA)，该机构将电子证书分配给任何需要它们的人。证书依赖于包含两个加密密钥（公钥和私钥）的非对称加密算法。

证书所有者可以向另一方提供证书作为身份证明。证书包含其所有者的公钥和其他详细信息，由可信任的CA 签名。使用此公钥加密的任何数据都只能使用相应的私钥进行解密，私钥由证书所有者持有。

例如，访问安全的(HTTPS) 电子商务站点时，该站点将向您的浏览器提供其证书。浏览器将根据其受信任CA 列表验证该证书，并使用其中包含的公钥创建加密的会话信息，该信息只能由发起服务器解密。然后，浏览器和服务器可以进行安全通信。

### 4.2.2对MySQL 服务器使用SSL

* SSL 使用要求：
* 系统必须支持yaSSL（已随MySQL 提供）或OpenSSL。
* 使用的MySQL 版本必须包含SSL 支持。
* 要获取安全连接以便使用MySQL 和SSL，必须首先执行

以下操作：

* 装入OpenSSL（如果使用的不是预编译的MySQL）。
* 为MySQL 配置SSL 支持。
* 确保mysql 数据库中的user 表包含与SSL 相关的列（ssl\_\* 和x509\_\*）。
* 使用--ssl 选项检查服务器二进制文件是否使用SSL 支持进行编译。
* 使用允许客户机使用SSL 的选项启动服务器。

为便于使用安全连接，MySQL 已附带了yaSSL。（MySQL 和yaSSL 采用相同的许可模

型，但是OpenSSL 使用Apache 样式的许可证。）

要获得OpenSSL，请访问http://www.openssl.org。使用OpenSSL 构建MySQL 需要

一个共享的OpenSSL 库；否则，将出现链接程序错误。

要将MySQL 源代码分发配置为使用SSL，请调用CMake：

shell> cmake . -DWITH\_SSL=bundled

* 该操作会将分发配置为使用附带的yaSSL 库。
* 要改用系统SSL 库，请将相应选项指定为-DWITH\_SSL=system。

如果user 表中没有与SSL 相关的列（以ssl\_\* 和x509\_\* 开头），则必须使用mysql\_upgrade 程序对其进行升级。

检查服务器二进制文件是否使用SSL 支持进行编译时，如果服务器不支持SSL，则会出

现错误：

shell> mysqld --ssl --help

060525 14:18:52 [ERROR] mysqld: unknown option '--ssl'

### 4.2.3使用SSL 启动MySQL 服务器

通过以下选项，使mysqld 服务器客户机可以使用SSL

进行连接：

* --ssl-ca：确定要使用的证书颁发机构(Certificate

Authority, CA) 证书（加密必需）

* --ssl-key：确定服务器公钥
* --ssl-cert：确定服务器私钥

shell> mysqld --ssl-ca=ca-cert.pem \

--ssl-cert=server-cert.pem \

--ssl-key=server-key.pem

可以向客户机发送--ssl-cert 并针对其正在使用的CA 证书进行验证。

### 4.2.4要求SSL 加密连接

* 使用GRANT 语句的REQUIRE SSL 选项可仅允许某个帐户使用SSL 加密连接：

GRANT ALL PRIVILEGES ON test.\*

TO 'root'@'localhost'

IDENTIFIED BY 'goodsecret' REQUIRE SSL;

* 通过使用--ssl-ca 选项启动mysql 客户机可获得加密连接。
* （可选）可以为X509 连接指定--ssl-key 和--ssl-cert 选项：

shell> mysql --ssl-ca=ca-cert.pem \

--ssl-cert=server-cert.pem \

--ssl-key=server-key.pem

要使用SSL 支持建立与MySQL 服务器的安全连接，必须指定的客户机选项取决于客户机所使用的用户帐户的SSL 要求。除基于用户名和口令的常见验证之外，MySQL 还可以检查X509 证书属性。要为MySQL 帐户指定与SSL 相关的选项，请使用GRANT 语句的REQUIRE 子句。

有很多方法可限制给定帐户的连接类型。在本课程中，只介绍三种选项：

* REQUIRE NONE：指示帐户没有SSL 或X509 要求。如果未指定与SSL 相关的REQUIRE 选项，则这是默认选项。如果用户名和口令有效，则允许未加密的连接。但是，如果客户机具有相应的证书和密钥文件，则客户机也可以通过指定一个选项来要求加密连接。即，客户机无需指定任何SSL 命令选项，在这种情况下，连接是未加密的。
* REQUIRE SSL：指示服务器仅允许帐户使用SSL 加密连接。
* REQUIRE X509：客户机必须具有有效的证书，但具体的证书、颁发者和主题无关紧要。唯一的要求是，应该可以使用其中一个CA 证书验证其签名。

### 4.2.5检查SSL 状态

* 使用have\_ssl 系统变量的值检查正在运行的

mysqld 服务器是否支持SSL：

mysql> SHOW VARIABLES LIKE 'have\_ssl';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| have\_ssl | YES |

+---------------+-------+

* 使用ssl\_cipher 状态变量的值检查当前服务器连接

是否使用了SSL：

mysql> SHOW STATUS LIKE 'ssl\_cipher';

+---------------+--------------------+

| Variable\_name | Value |

+---------------+--------------------+

| ssl\_cipher | DHE-RSA-AES256-SHA |

+---------------+--------------------+

如果have\_ssl 的值为YES，则服务器支持SSL 连接。如果值为DISABLED，则服务器支持SSL 连接，但在启动时未提供相应的--ssl-\* 选项。

对于mysql 客户机，可以使用STATUS 或\s 命令并检查SSL 行：

mysql> \s

...

SSL: Cipher in use is DHE-RSA-AES256-SHA

...

### 4.2.6使用SSL 的优点和缺点

|  |  |
| --- | --- |
| 优点 | 缺点 |
| 提高了有需求的应用程序的安全性 | 占用大量CPU资源 |
| 可以针对单个连接启用 | 降低了客户机/服务器协议的速度 |
| 可用于复制操作 | 可能会推迟其他SQL任务 |

## 4.3与MySQL 的安全远程连接

* MySQL 支持与远程MySQL 服务器的SSH（secure shell，安全shell） 连接。这一功能要求：
* 客户机上存在SSH 客户机
* 通过SSH 隧道进行从客户机到服务器的端口转发
* 具有SSH 客户机的计算机上存在客户机应用程序
* 设置完成后，将有一个本地端口托管到MySQL 的SSH 连接并使用SSH 进行加密。

## 4.4文件系统安全

MySQL 安装（目录和文件）风险预防任务：

* 更改所有权和访问权限后，再启动服务器。
* 将多用户系统的所有权设置为具有管理特权的帐户。
* 将与MySQL 相关的目录和文件以及user和group表所有权设置为mysql，其中包括：
* MySQL 程序
* 数据库目录和文件
* 日志、状态和配置文件
* 不要在保护文件之前设置口令。否则，将允许未经授权的用户替换文件。
* 设置一个专用于MySQL管理的账户。

请勿向任何人（MySQL root 帐户除外）授予对mysql 数据库中的user 表的访问权限。将MySQL 置于防火墙后或隔离区(demilitarized zone, DMZ) 中。

仅当服务器正在运行时，才可以设置MySQL root 帐户的口令。因此，启动服务器并设置口令之前，请采取任何必要的措施来保护文件系统中与MySQL 相关的部分。如果在保护相应文件之前设置了口令，则有权直接访问服务器主机上文件系统的用户可能会替换这些文件。这会危害MySQL 安装并撤消口令设置的效果。

对于Linux之类的多用户系统，请将MySQL安装的所有组件的所有权设置为具有正确管理特权的专用登录帐户。这将保护安装免受不负责数据库管理的用户的访问。设置此帐户的另外一个好处是，可以使用该帐户运行MySQL 服务器，而不是从Linux root 帐户运行服务器。具有root 登录帐户特权的服务器拥有不必要的文件系统访问权限，从而成为一个安全风险。

**文件系统安全设置：**

* Mysql配置文件权限属性 # *chmod 600 /etc/my.cnf*

MHA 配置文件权限属性# *chmod 600 /etc/app1.cnf*

Atlas 配置文件权限属性 # *chmod 600 /etc/test.cnf*

* Mysql相关的安装目录，数据目录所属组和属主都改为mysql

# *chown mysql:mysql /mydata/mysql*

## 4.5用户安全设置

1. 确保任何可以访问数据的用户都有一个比较复杂内容的密码
2. Mysql正式启用之前，将root用户加密码，更改为非默认用户并只允许本地登录
3. 尽可能为每一个账户限定一定范围的可访问主机
4. 用户权限最小化 insert\select\update\delete\select view\call pro\call fun
5. 定期修改所有用户密码