MySQL 安全配置

1 安全策略

1.1 管理意义上的数据安全

访问 MySQL 数据库必须首先访问数据库的某个权限、即以某个权限模式用户的身份登录，大部分的安全管理主要通过模式用户的权限来实现。

MySQL 的相关权限信息主要存放在 grant tables 的系统表中，即 mysql.User(全局级别权限) 、 mysql.db (数据库级别权限)、 mysql.Host(数据库级别权限) 、mysql.table_priv(表级别权限) 和、 mysql.column_pr(列级别权限)表中， MySQL 启动时装入内存。应尽量使用 GRANT 、REVOKE 、CREATE USER 及 DROP USER 来进行用户和权限的变更操作。

```
    如 :GRANT SELECT.UPDATE,DELETE,INSERT,EXECUTE ON test_
shop.* TO 'test_guest'@'localhost';
```

查看某用户权限，如

```
    SHOW GRANTS FOR 'test_guest'@'localhost';
```

1.2 防范故障角度的数据安全

数据文件是操作系统级的对象，因此一般来讲具有相当的脆弱性、而且依赖于操作系统的性能特点。由于磁盘介质的因素、一个大的数据文件上个别数据块的损坏可能导致整个数据文件的不可用，这对一个系统来说是灾难性的，而且大的表空间或数据文件的恢复是困难和耗时的。

巨大对象的分区在性能角度之外也有安全的因素，当磁盘错误使一个巨大表中一个单独的数据块不能读写时可能导致整个表不可用，必须恢复包含该表的整个表空间。

考虑到数据仓库问题。可以进行以下操作：

对数据量大且不进行写操作的表，使用 myisampack 工具，生成压缩、只读 MyISAM 表。可以压缩 40% - 50% 的表文件空间。具体操作如下：

A 压缩文件： >myisampack ../data/music_shop/ 表名 .MYI

B 重建索引： >myisamchk -rq --sort-index --analyze../data/test_shop/ 表名 .MYI

C 强制 mysqld 使用新表： > mysqladmin flush-tables

如果要进行写操作，可以解压缩一个压缩的表，恢复原有状态，使用 myisamchk 。 如： myisamchk --unpack ../data/music_shop/ 表名 .MYI

最后，系统上线后，随着数据量的增加，会发现数据目录下的磁盘空间越来越下，造成安全隐患。可以采取两种措施。一种针对 MyISAM 存储引擎的表，在建表时分别指定数据目录和索引目录到不同的磁盘空间，而默认会同时放在数据目录下。另外一种针对 InnoDB 存储引擎的表，因为数据文件和索引文件在一起的，所以无法将它们分离。当磁盘空间不足时，可以增加一个新的数据文件，这个文件放在有充足空间的磁盘上。具体请查阅参数 innodb_data_file_path 设置。

1.3 容灾与备份机制

建立主从数据库集群，采用 MySQL 复制

MySQL 复制的优点：

1 如果主服务器出现问题，可以快速切换到从服务器;

2 可以在从服务器上执行查询操作，降低主服务器的访问压力;

3 可以在从服务器上执行备份，以避免备份期间影响主服务器的;

应注意的问题：

由于实现的是异步的复制，所以主从服务器之间存在一定的差距。在从服务器上进行的查询操作要考虑到这些数据的差异，一般只有对实时性要求不高的数据可以通过从服务器查询。

定期备份文件与数据，通过各种方式保存文件与数据。

以下是几点防范的措施：

制定一份数据库备份 / 恢复计划，并对计划进行仔细测试。

启动数据库服务器的二进制变更日志，该功能的系统开销很小（约为 1%），二进制日志包含备份后进行的所有更新，我们没有理由不这样做。(log-bin=file，file 可以不指定)

定期检查数据表，防范于未燃。

定期对备份文件进行备份，以防备份文件失效。

把 MySQL 的数据目录和备份文件分别放到两个不同的驱动器中，以平衡磁盘 I/O 和增加数据的安全。

2 安全隐患

2.1 正确设置目录权限

设置目录权限的原则是软件和数据分开，具体如下：

1. 将 mysql 安装在单独的用户下

2. 安装时，以 root 用户进行安装，mysql 的软件默认都为 root 权限

3. 安装完毕后，将数据目录权限设置为实际运行 mysql 的用户权限，比如：

Chown –R mysql:mysql /home/mysql/data

2.2 尽量避免以 root 权限运行 mysql

将 4.1 的目录权限设置完毕后，启动、停止 mysql 以及日常的维护工作都可以在 mysql 用户下进行，没有必要 su 到 root 后再用—user=mysql 来启动和关闭 mysql，这样就没有必要授权维护人员 root 权限，而且最重要的一定是因为任何具有 FILE 权限的用户能够用 root 创建文件。

2.3 删除匿名账号

有些版本的 MySQL 安装完之后会安装一个空账号(User = '')，此账号对 test 数据库有完全权限，为避免此账号登陆后，建立大表，占用磁盘空间，影响系统安全，建议删除

```
drop user ''@'localhost';
drop user ''@' localhost.localdomain’;
```

2.4 给 root 账号设置口令

建议以一句话的拼音为口令。如

```
set PASSWORD=PASSWORD('woshiyitiaoyu');
```

并且限定只能通过 localhost 访问。

2.5 只授予账号必须的权限

```
如： Grant select,insert,update,delete on tablename to 'username'@'hostname';
```

2.6 除 root 外，任何用户不应有 mysql 库 user 表的存取权限

如果拥有 mysql 库中 user 表的存取权限(select、update、insert、delete)，就可以轻易的增加、修改、删除其他的用户权限，造成系统的安全隐患。

如：

```
    use mysql;delete from db where user<>'root' and db='my
sql';
```

2.7 不要把 file 、 process 、或 super 权限授予管理员以外的账号

会产生保密信息外泄，查看管理员执行的动作，普通用户执行 kill 命令等严重的安全隐患。

FILE 权限可以被滥用于将服务器主机上 MySQL 能读取的任何文件读入到数据库表中。包括任何人可读的文件和服务器数据目录中的文件。可以使用 SELECT 访问数据库表，然后将其内容传输到客户端上。不要向非管理用户授予 FILE 权限。

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!为了更加安全，由 SELECT ... INTO OUTFILE 生成的所有文件对每个人是可写的，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 用户可读取或访问的文件。使用该权限，你可以将任何文件读入数据库表。这可能被滥用，例如，通过使用 LOADDATA 装载“/etc/passwd”进一个数据库表，然后能用 SELECT 显示它。PROCESS 权限能被用来察看当前执行的查询的明文文本，包括设定或改变密码的查询。

SUPER 权限能用来终止其它用户或更改服务器的操作方式。比如 kill 进程不要将 PROCESS 或 SUPER 权限授给非管理用户。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 UPDATE user SETpassword=PASSWORD('not_secure')查询，被允许执行那个命令的任何用户可能看得到

2.8 LOAD DATA LOCAL 带来的安全问题

由 MySQL 服务器启动文件从客户端向服务器主机的传输。理论上，打过补丁的服务器可以告诉客户端程序传输服务器选择的文件，而不是客户用 LOAD DATA 语句指定的文件。这样服务器可以访问客户端上客户有读访问权限的任何文件。

在 Web 环境中，客户从 Web 服务器连接，用户可以使用 LOAD DATA LOCAL 来读取 Web 服务器进程有读访问权限的任何文件(假定用户可以运行 SQL 服务器的任何命令)。在这种环境中，MySQL 服务器的客户实际上是 Web 服务器，而不是连接 Web 服务器的用户运行的程序。

解决方法：

可以用--local-infile=0 选项启动 mysqld 从服务器端禁用所有 LOAD DATA LOCAL 命令。

对于 mysql 命令行客户端，可以通过指定--local-infile[=1]选项启用 LOAD DATA LOCAL，或通过--local-infile=0 选项禁用。类似地，对于 mysqlimport，--local or -L 选项启用本地数据文件装载。在任何情况下，成功进行本地装载需要服务器启用相关选项。

2.9 使用 MERGE 存储引擎潜藏的安全漏洞

Merge 表在某些版本中可能存在以下安全漏洞：

用户 A 赋予表 T 的权限给用户 B

用户 B 创建一个包含 T 的 merge 表，做各种操作

用户 A 收回对 T 的权限

安全隐患：用户 B 通过 merge 表仍然可以访问表 A 中的数据

2.10 尽量避免通过 symlinks 访问表

不要允许使用表的符号链接。(可以用--skip-symbolic-links 选项禁用)。如果你用 root 运行 mysqld 则特别重要，因为任何对服务器的数据目录有写访问权限的人则能够删除系统中的任何文件!

2.11 防止 DNS 欺骗

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。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来创建 授权表条目!

2.12 DROP TABLE 命令并不收回以前的相关访问授权

drop 表的时候，其他用户对此表的权限并没有被收回，这样导致重新创建同名的表时，以前其他用户对此表的权限会自动赋予，导致权限外流。因此，要在删除表时，同时取消其他用户在此表上的相应权限。

2.13 REVOKE 命令漏洞

```
    grant all privileges on *.* to guest@localhost; 后

    revoke all privileges on *.* from guest@localhost; 不起作用，必须针对每个数据单独使用 revoke
```

2.14 如果可能，给所有用户加上访问 IP 限制

给所有用户加上 ip 限制将拒绝所有未知的主机进行的连接，保证只有受信任的主机才可以进行连接。例如：

```
   Grant select on dbname.* to 'username'@'ip' identified
 by 'passwd';
```

2.15 严格控制操作系统帐号和权限

在数据库服务器上要严格控制操作系统的帐号和权限，比如：锁定 mysql 用户，其他任何用户都采取独立的帐号登陆，管理员通过普通用户管理 mysql; 或者通过 root su 到 mysql 用户下进行管理。

禁止修改 mysql 用户下的任何资源。

2.16 增加防火墙

购买防火墙。这样可以保护你防范各种软件中至少 50%的各种类型的攻击。把 MySQL 放到防火墙后或隔离区(DMZ)

2.17 严格模式

```
   sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_E
NGINE_SUBSTITUTION"
```

2.18 限制 MYSQL 的访问目录

```
   --chroot
```

2.19 防止在连接 MYSQL 是使用 TCP/IP 套接字

```
   --skip-networking
```

2.20 防止连接到 MYSQL 数据库时使用主机名

```
--skip-name-resolve
```

2.21 防止没有 SHOW DATABASES 权限的用户使用此命令

```
--skip-show-database
```

2.22 如果对 user 表没有 INSERT 权限，可以防止这些用户通过 GRANT 命令创建用户

```
--safe-user-create
```

3 其他安全配置

MySQL 本身带有一些选项，适当的使用这些选项将会使数据库更加安全。

3.1 使用 skip-network

在网络上不允许 TCP/IP 连接，所有到数据库的连接必须由命名管道 (Named Pipes) 或共享内存(Shared Memory) 或 UNIX 套接字 SOCKET 文件进行。这个选项适合应用和数据库共用一台服务器的情况，其他客户端将无法通过网络远程访问数据库，大大增强了数据库的安全性，但同时也带来了管理维护上的不方便。 MySQL 仅能通过命名管道或共享内存（在 widows 中）或 Unix 套接字文件（在 Unix 系统中）来和客户端连接交互。以下为配置实例：

```
skip-networking
```

-S 是 --socket 的简写形式，如： -s /tmp/mysql.sock，而其值必须和服务端设置的相同。

--protocol 是严格指定连接类型，如果一些设置使用默认值时，如 windows 下服务器端设置--socket=mysql(mysql 是默认值)，在连接时指定 --protocol=pipe 后 --socket=mysql 可省略指定。

1. 命名管道

只适合在 Windows 系统下用来连接本机的 MySQL ，性能可比一般的 TCP/IP 方式提升 30%~50%。

服务端设置要求

```
enable-named-pipe #或 named_pipe=ON
socket=MySQL
```

客户端连接

```
mysql --protocol=pipe --socket=mysql
```

2. 共享内存

4.1 版本后，mysql 对 windows 系统还提供了共享内存方式的连接

服务端设置要求

```
shared-memory=ON
shared_memory_base_name=MYSQL
```

客户端连接

```
mysql --protocol=memory --shared-memory-base-name=mysq
l
```

3. UNIX 套接字

linux 和 unix 环境下，可以使用 unix 域套接字，来连接同在一台机器上的 mysql;

服务端设置要求

```
socket=/tmp/mysql.sock
```

客户端连接

```
mysql --protocol=socket --socket=/tmp/mysql.sock
```

3.2 allow-suspicious-udfs

该选项控制是否可以载入主函数只有 xxx 符的用户定义函数。默认情况下，该选项被关闭，并且只能载入至少有辅助符的 UDF。这样可以防止从未包含合法 UDF 的共享对象文件载入函数。

3.3 old-passwords

强制服务器为新密码生成短(pre-4.1)密码哈希。当服务器必须支持旧版本客户端程序时，为了保证兼容性这很有用。

3.4 safe-user-create

如果启用， 用户不能用 GRANT 语句创建新用户，除非用户有 mysql.user 表的 INSERT 权限。如果你想让用户具有授权权限来创建新用户，你应给用户授予下面的权限：

```
mysql> GRANT INSERT(user) ON mysql.user TO 'user_name
'@'host_name';
```

这样确保用户不能直接更改权限列，必须使用 GRANT 语句给其它用户授予该权限。

3.5 secure-auth

不允许鉴定有旧(pre-4.1)密码的账户

3.6 skip-grant-tables

这个选项导致服务器根本不使用权限系统。这给每个人以完全访问所有的数据库的权力!(通过执行 mysqladmin flush-privileges 或 mysqladmin reload 命令，或执行 FLUSH PRIVILEGES 语句，你能告诉一个正在运行的服务器再次开始使用授权表 。)

3.7 skip-show-database

使用该选项，只允许有 SHOW DATABASES 权限的用户执行 SHOW DATABASES 语句，该语句显示所有数据库名。不使用该选项，允许所有用户执行 SHOW DATABASES，但只显示用户有 SHOW DATABASES 权限或部分数据库权限的数据库名。请注意全局权限指数据库的权限。

3.8 使用 SSL

SSL（Secure Socket Layer 安全套接字)是一种安全协议，最初由 Netscape 公司所开发，用以保障在 Internet 上数据传输的安全，利用数据加密技术，可确保数据在网络上的传输过程中不会被截取。

应用场景，在主从数据库复制中使用，提供以下服务保障。

a) 认证用户和服务器，确保数据发送到正确的客户和服务器。

b) 加密数据以防止数据中途被窃取。

c) 维护数据的完整性，确保数据在传输过程中不被破坏。

在 MySql 中使用 SSL 进行安全传输，需要在命令行或选项文件中设置‘SSL’ 选项。下面以命令行为例，进行安装介绍。

A. 安装证书管理工具

a) 所需部件 Win32OpenSSL-0_9_8g.exe ，可从网上下载

b) 安装 双击 Win32OpenSSL-0_9_8g.exe 按提示进行安装。安装在 C:\OpenSSL 目录下

c) 在 C:\OpenSSL\bin 目录下创建 root ，server ，client 三个子路径

d) 在创建证书时输入的用户名，密码请妥善保存

B. 创建根证书，并采用自签名签署它

a) 创建私钥 进入 DOS 窗口，进入 C:\OpenSSL\bin 路径，然后输入 openssl genrsa -out root/root-key.pem 1024 命令，按 Enter 键。

b) 创建证书请求 继续输入 openssl req -new -out root/root-req.csr -key root/root-key.pem ，然后按 Enter 键，要求输入一系列信息，可根据实际情况输入，但是 CommonName ：一定要输入 root

c) 自签署根证书 继续输入 openssl x509 -req -in root/root-req.csr -out root/root-cert.pem -signkey root/root-key.pem -days 3650 ，然后按 Enter 键

d) 查看根证书内容 要先进入证书所在路径 例：C:\OpenSSL\bin\root ，然后输入 keytool -printcert -file root-cert.pem，然后按 Enter 键。

C. 创建服务器证书，并采用根证书签署它

a) 创建私钥 进入 DOS 窗口，进入 C:\OpenSSL\bin 路径，然后输入 openssl genrsa -out server/server-key.pem 1024 命令，按 Enter 键。

b) 创建证书请求 继续输入 openssl req -new -out server/server-req.csr -key server/server-key.pem ，然后按 Enter 键，要求输入一系列信息，可根据

实际情况输入，但是 CommonName ：一定要输入 localhost 或服务器的域名(存在域名情况下)。

c) 签署服务器证书 继续输入 openssl x509 -req -in server/server-req.csr -out server/server-cert.pem -signkey server/server-key.pem -CA root/root-cert.pem -CAkey root/root-key.pem -CAcreateserial -days 3650 ，然后按 Enter 键。

d) 查看服务器证书内容 要先进入证书所在路径 例：C:\OpenSSL\bin\server ，然后输入 keytool -printcert -file server-cert.pem ，然后按 Enter 键。

D. 创建客户证书，并采用根证书签署它

a) 创建私钥 进入 DOS 窗口，进入 C:\OpenSSL\bin 路径，然后输入 openssl genrsa -out client/client-key.pem 1024 命令，按 Enter 键。

b) 创建证书请求 继续输入 openssl req -new -out client/client-req.csr -key client/client-key.pem ，然后按 Enter 键，要求输入一系列信息，可根据实际情况输入，CommonName ：输入用户 ID 。

c) 签署客户证书 继续输入 openssl x509 -req -in client/client-req.csr -out client/client-cert.pem -signkey client/client-key.pem -CA root/root-cert.pem -CAkey root/root-key.pem -CAcreateserial -days 3650 ，然后按 Enter 键。

d) 查看客户证书内容 要先进入证书所在路径 例：C:\OpenSSL\bin\client ，然后输入 keytool -printcert -file client-cert.pem ，然后按 Enter 键。

完成以上步骤后，将所生成的证书 root 、server 和 client 文件夹，拷到 C:\mysll 目录下。 至此，已部署完在启动服务器时所用的有关选项指明证书文件和密钥文件。在建立加密连接前，要准备三个文件，一个 CA 证书，是由可

信赖第三方出具的证书，用来验证客户端和服务器端提供的证书。 CA 证书可向商业机构购买，也可自行生成。第二个文件是证书文件，用于在连接时向对方证明自己身份的文件。第三个文件是密钥文件，用来对在加密连接上传输数据的加密和解密。 MySQL 服务器端的证书文件和密钥文件必须首先安装，在 myssl 目录里的几个文件：root-cert.pem(CA 证书）， server-cert.pem(服务器证书）， server-key.pem(服务器公共密钥）。

在主数据库创建从数据库操作所用的用户，并指定必须用 SLL 认证。

```
    CREATE USER 'test_guest'@'localhost' IDENTIFIED BY '12
34';

    GRANT ALL PRIVILEGES ON music_shop.* TO 'test_guest '@
'10.12.1.42' REQUIRE ssl;
```

关闭主数据库

```
    >mysqladmin -uroot shutdown
```

重启服务器，使配置生效。

```
    >mysqld--ssl-ca=C:\myssl\server\root-cert.pem --ssl-ce
rt=C:\myssl\server\server-cert.pem --ssl-key=C:\myssl\ser
ver\server-key.pem
```

用从数据库客户程序建立加密连接。

```
    >mysql -u test_guest --ssl-ca=C:\myssl\client\root-cer
t.pem --ssl-cert=C:\myssl\client\client-cert.pem --ssl-ke
y=C:\myssl\client\client-key.pem
```

配置完成后，调用 mysql 程序运行 \s 或 SHOW STATUS LIKE ‘SSL%’ 命令，如果看到 SSL: 的信息行就说明是加密连接了。如果把 SSL 相关的配置

写进选项文件，则默认是加密连接的。也可用 mysql 程序的 --skip-ssl 选项取消加密连接。