

版本加固:

满足业务正常运行的前提下, 安装新版本, 修补漏洞检查方法  
检查方法

>查看 mysql 版本信息:mysql> select version();

加固方法: 安装最新版 mysql, <http://www.mysql.com>

弱口令

安全基线项说明: 确保数据库不存在弱口令, 提高数据库安全性

检查方法:

use mysql

select user,password from mysql.user

将密码 hash 导入 cain 软件破解

加固方法如要修改密码, 执行如下命令

首先以 root 用户登录

mysql>use mysql;

mysql>update user set password=password ("复杂的新密码") where user='test';

mysql> flush privileges;

是否存在匿名账户

安全基线项说明: 确保数据库不存在匿名账户工提高数据库安全性

检查方法: 检查匿名帐户是否存在

mysql>use mysql;

mysql> select user,password from mysql.user;

存在 user 和 password 字段均为空的行

加固方法:

删除匿名帐户:

mysql>use mysql;

mysql>delete from user where user='',

mysql>flush privileges;

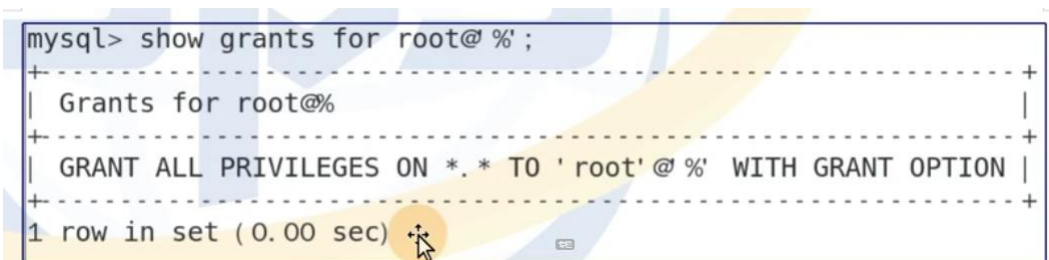
合理设置权限

安全基线项说明: 合理设置用户权限, 防止权限滥用

检查方法:

查看用户权限:show grants for test@localhost;

mysql>show grants for root@'%';



```
mysql> show grants for root@ '%';
+-----+
| Grants for root@% |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION |
+-----+
1 row in set (0.00 sec)
```

加固方法: 一般应用用户建议授予最小权限

方法:grant 权限 1 权限 2..权限 n on 数据库名称.表名称 to 用户名@用户地址

例如：

```
grant select,insert,delete on db.table to test@localhost;
```

合理设置文件权限：

安全基线项说明：合理设置数据库文件权限，防止否授权访问或篡改

检查方法：确保重要的数据库文件没有任意可写权限或任意可读权限检查是否有不恰当的授权文件：

```
>#ls -al .mysql_history .bash_history 应为 600 权限
```

```
>#ls -al /etc/my.cnf 应为 644 权限
```

```
>#find /-name .MYD xargs ls -al 应为 600 权限
```

```
>#find /-name .MYI | xargs ls -al 应为 600 权限
```

```
>#find /-name frm | xargs ls -al 应为 600 权限
```

加固方法：保护数据库文件，授予恰当的权限：

```
#chmod 600 .mysql_history .bash_hatory
```

```
#chmod 600*.MYD*.MYI*.frm
```

```
#chmod o-rw /etc/my.cnf
```

日志审核：

安全基线项说明：合理设置日志审核，保证安全事件发生可查看日志记录

检查方法：查看 my.cnf 或 my.ini 文件，查看是否包含如下设置

```
[mysqld]
```

```
log = filename
```

加固方法：在 mysql 的安装目录下，打开 my.cnf 或 my.ini，在 **【mysqld】**

后面加上如下的参数，取消注释并配置日志文件，保存后重启 mysql 服务就行了。

```
#Enter a namefor the binary log. Otherwise a default name will be used.
```

```
#log-bin=
```

```
#Enter a name for the query log file. Otherwise a default name will be used.
```

```
#log=
```

```
#Enter a name for the error log file. Otherwise a default name will be used.
```

```
log-error=
```

```
#Enter a name for the update log file. Otherwise a default name will be used.
```

```
#log-update=
```

（去掉全面的#，加上路径）

运行账号：

安全基线项目名称：Mysqld 服务以普通用户运行，防止数据库高权限被利用

检查方法：检查进程属主和运行参数是否包含-user=mysql 类似语句：

```
ps -ef | grep mysqld
```

```
grep -i user /etc/my.cnf
```

加固方法 vi 编辑/etc/my.cnf，设置如下：

```
[mysql.server]
```

```
user=mysql
```

可信 IP 地址控制

口安全基线项说明：只允许可信任的 ip 访问数据库，降低数据库风险

口检查方法：查看可访问数据库的 ip 和 ip 对应的账号

```
mysql> select user,host from mysql.user;
```

加固方法

```
mysql> GRANT ALL PRIVILEGES ON ** TO '可信任用户'@'可信任 ip 地址' IDENTIFIED BY '可信任用户密码' WITH GRANT OPTION;
```

连接数限制：

安全基线项说明：根据业务需求设置数据库最大连接数检查方法：查看 MySQL

配置文件：my.cnf 或者是 my.ini

在【mysqld】段查看最大连接数配置：

max\_connections=1000 查看最大连接数

```
mysql> show variables like 'max connections';
```

加固方法：

编辑 MySQL 配置文件：my.cnf 或者是 my.ini

在【mysqld】配置段添加：

```
max_connections = 1000
```

保存，重启 MySQL 服务。

更严格的一些基线要求：

禁止远程连接数据库：

安全基线项说明：在命令行 netstat-ant 下看到，默认的 3306 端口是打开的，此时打开了 mysqld 的网络监听，允许用户远程通过帐号密码连接数据库，默认情况是允许远程连接数据的。

检查方法

```
show variables like "%skip_networking%";
```

```
show variables like "%bind_address%";
```

加固方法：

为了禁止该功能，启动 skip\_networking，不监听 sql 的任何 TCP/IP 的连接，切断远程访问的权利，保证安全性。

```
# vim /etc/my.cnf
```

```
[mysqld]
```

```
bind_address=127.0.0.1
```

```
skip_networking=1
```

假如需要远程管理数据库，可通过安装 PhpMyadmin 来实现。假如确实需要远程连接数据库，至少修改默认的监听端口，同时添加防火墙规则，只允许可信任的网络的 mysql 监听端口的数据通过。

改变默认 mysql 管理员账号：

安全基线项说明：系统 mysql 的管理员名称是 root，而一般情况下，数据库管理员都没进行修改，这一定程度上对系统用户穷举的恶意行为提供了便利，此时修改为复杂的用户名，请不要再设定为 admin 或者 administrator 的形式，因为它们也在易猜的用户字典中。

加固方法

改成不易被猜到的用户名：

```
mysql> update user set user="newroot" where user="root";  
mysql> flush privileges;
```

删除默认数据库：

安全基线项说明：MySQL 有的版本初始化后会自动生成空用户和 test 库，这会对数据库的安全构成威胁，有必要全部删除，最后的状态只保留单个 root 即可，当然以后根据需要增加用户和数据库。

加固方法：

```
#mysql> show databases;  
#mysql> drop database test; //删除数据库 test
```

命令历史记录保护：

安全基线项说明：数据库相关的 shell 操作命令都会分别记录在 .bash\_history，如果这些文件不慎被读取，会导致数据库密码和数据库结构等信息泄露，而登陆数据库后的操作将记录在 .mysql\_history 文件中，如果使用 update 表信息来修改数据库用户密码的话，也会被读取密码，

加固方法：

需要删除这两个文件获奖者将文件置空

同时在进行登陆或备份数据库等与密码相关操作时，应该使用 -p 参数加入提示输入密码后，隐式输入密码

```
# rm.bash_history.mysql_history //删除历史记录  
#ln -s /dev/null.bash_history //将 shell 记录文件置空  
#ln -s /dev/null.mysql_history //将 mysql 记录文件置空
```

禁止 mysql 对本地文件存取：

安全基线项说明：在 mysql 中，提供对本地文件的读取，使用的是 load data localinfile 命令默认在 5.0 版本中，该选项是默认打开的，该操作令会利用 MySQL 把本地文件读到数据库中，然后用户就可以非法获取敏感信息了，假如你不需要读取本地文件，请务必关闭。

加固方法：

--local-infile=0 选项启动 mysqld 从服务器端禁用所有 LOAD DATA LOCAL 命令，假如需要获取本地文件，需要打开，但是建议关闭。

可以在 my.cnf 中添加 local-infile=0，或者加参数 local-infile=0

重新启动 mysql。