15. Windows 注册表访问命令详解

以下这些选项可以用来访问后端数据库管理系统 Windows 注册表。

--reg-read 读一个 Windows 注册表项值；

--reg-add 写一个 Windows 注册表项值数据；

--reg-del 删除 Windows 注册表键值；

--reg-key=REGKEY Windows 注册表键；

--reg-value=REGVAL Windows 注册表项值；

--reg-data=REGDATA Windows 注册表键值数据；

--reg-type=REGTYPE Windows 注册表项值类型。

16. 一般工作参数的命令详解

以下这些选项可以用来设置一些一般的工作参数。

-t TRAFFICFILE 记录所有 HTTP 流量到一个文本文件中；

-s SESSIONFILE 保存和恢复检索会话文件的所有数据；

--flush-session 刷新当前目标的会话文件；

--fresh-queries 忽略在会话文件中存储的查询结果；

--eta 显示每个输出的预计到达时间；

--update 更新 sqlmap；

--save file 保存选项到 INI 配置文件；

--batch 从不询问用户输入，使用所有默认配置。

17. Miscellaneous（杂项）命令详解

--beep 发现 SQL 注入时提醒；

--check-payload IDS 对注入 payloads 的检测测试；

--cleanup SqlMap 具体的 UDF 和表清理 DBMS；

--forms 对目标 URL 的解析和测试形式；

--gpage=GOOGLEPAGE 从指定的页码使用谷歌 dork 结果；

--page-rank Google dork 结果显示网页排名（PR）；

--parse-errors 从响应页面解析数据库管理系统的错误消息；

--replicate 复制转储的数据到一个 sqlite3 数据库；

--tor 使用默认的 Tor（Vidalia/ Privoxy/ Polipo）代理地址；

--wizard 给初级用户的简单向导界面。

任务4　MySQL 数据库加固技术应用

【任务描述】

某公司的很多服务都用了 MySQL 服务器，但是 MySQL 服务器是个开源的服务器，有很多已知的漏洞，如果没有很好地进行补丁和安全配置，将会带来很大灾难。

【任务分析】

本任务需要在目标主机上开启 Web 服务、MySQL 服务、PHP 服务和 Nmap 服务。对目标主机上的 MySQL 服务器进行加固，以确保数据库服务器的安全。

扫描全能王 创建

【任务实施】

（1）修改 root 用户密码，删除空密码。

缺省安装的 MySQL 的 root 用户是空密码的，为了安全起见，必须修改为强密码。所谓强密码，是至少 8 位，由字母、数字和符号组成的不规律密码。使用 MySQL 自带的命令 mysqladmin 修改 root 密码，同时也可以登录数据库，修改数据库 MySQL 下 user 表的字段内容。修改方法如图 5－37、图 5－38 和图 5－39 所示。

```
# /usr/local/mysql/bin/mysqladmin -u root password " upassword" //使用 mysqladmin
```

```
[root@promote 桌面]# mysqladmin -u root password "upassword"
[root@promote 桌面]#
```

图 5－37　使用 mysqladmin 修改用户密码

```
#mysql >use mysql;
#mysql >update user set password = password ('123456') where user ='root';
```

```
mysql> update user set password=password('123456') where user='root';
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql>
```

图 5－38　更新 user 表中的字段来修改用户密码

```
#mysql >flush privileges; //强制刷新内存授权表，否则用的还是在内存缓冲的密码
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

图 5－39　强制刷新内存授权表

（2）删除默认数据库和数据库用户。一般情况下，MySQL 数据库安装在本地，并且也只需要本地的 php 脚本对 MySQL 进行读取。MySQL 初始化后，会自动生成空用户和 test 库进行安装的测试，这会对数据库的安全构成威胁，有必要全部删除，最后只保留单个 root 即可。执行过程如图 5－40、图 5－41 和图 5－42 所示。当然，以后根据需要可以随时增加用户和数据库。

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

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> drop database test;
Query OK, 0 rows affected (0.08 sec)

mysql>
```

图 5－40　删除数据库

```
#use mysql;
#delete from db; //删除存放数据库的表信息，因为还没有数据库信息
```


扫描全能王 创建

```
#mysql >delete from user where not (user ='root'); //删除初始非 root 的用户
```

```
mysql> use mysql;
Database changed
mysql> delete from db;
Query OK, 2 rows affected (0.00 sec)

mysql> delete from user where not (user='root');
Query OK, 2 rows affected (0.00 sec)
```

图 5－41　删除非 root 用户信息

```
#mysql >delete from user where user ='root' and password = '';
//删除空密码的 root，尽量重复操作 Query OK, 2 rows affected (0.00 sec)
#mysql >flush privileges; //强制刷新内存授权表
```

```
mysql> delete from user where user='root' and password='';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

图 5－42　删除空密码的 root 用户

（3）改变默认 MySQL 管理员的名称。将系统的默认管理员 root 改为 admin，以防被列举。执行过程如图 5－43 所示。# select host,user from user;

```
mysql> delete from user where user='';
Query OK, 1 row affected (0.00 sec)

mysql> update user set user='admin' where user='root';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select host,user from user;
+-----------+------------------+
| host      | user             |
+-----------+------------------+
| %         | admin            |
| 127.0.0.1 | admin            |
| localhost | admin            |
| localhost | debian-sys-maint |
+-----------+------------------+
4 rows in set (0.00 sec)
```

图 5－43　修改管理员账户名称

（4）进入目标主机，加固 MySQL 服务器，使所有的访问能被审计。通过对 mysqld 的启动项进行加固（命令如图 5－44 所示），修改配置文件参数，如图 5－45 所示。

```
# find / -name my.cnf
/etc/mysql/my.cnf
# vi /etc/mysql/my.cnf
#
```

图 5－44　查找并打开配置文件

（5）配置 Linux 操作系统的防火墙，允许 MySQL 服务器能够被访问，要求规则中只包含端口项，对防火墙规则列表进行设置操作，如图 5－46 所示。这里需要知道 MySQL 的端口号是 3306。

（6）进入目标主机连接本地 MySQL 数据库，如图 5－47 所示；查看所有用户及权限，找到可以从任何 IP 地址访问的用户，结果如图 5－48 所示。

（7）对数据库中存在的漏洞进行加固，设定该用户只能从公司 PC－1 访问，用 grants 命令进行管理，操作命令如图 5－49 所示。

扫描全能王 创建

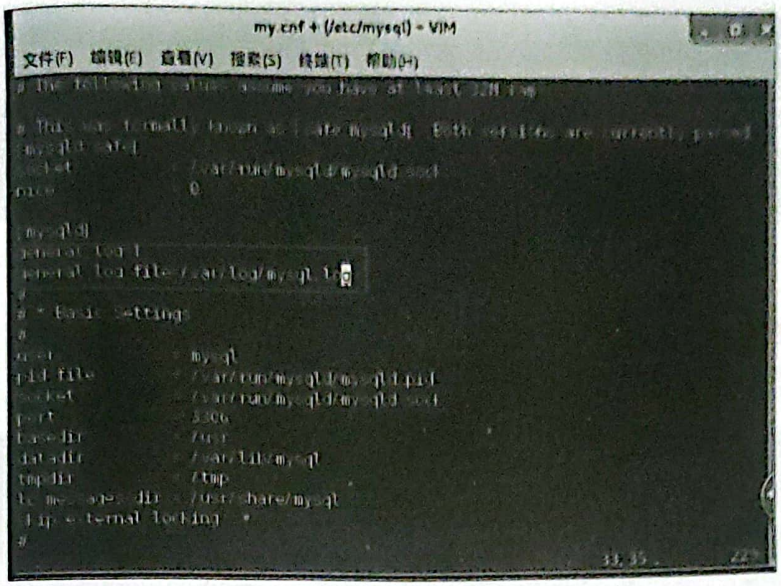

图5-45　修改配置文件参数

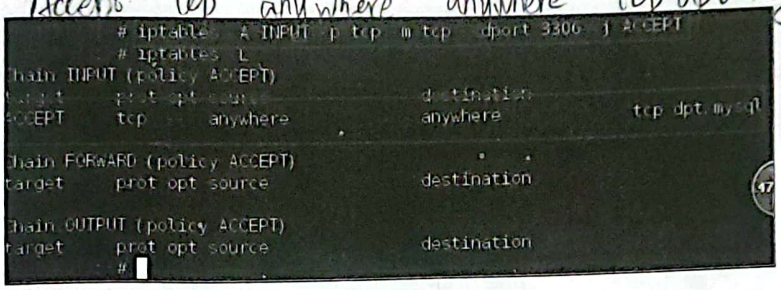

图5-46　配置Linux防火墙

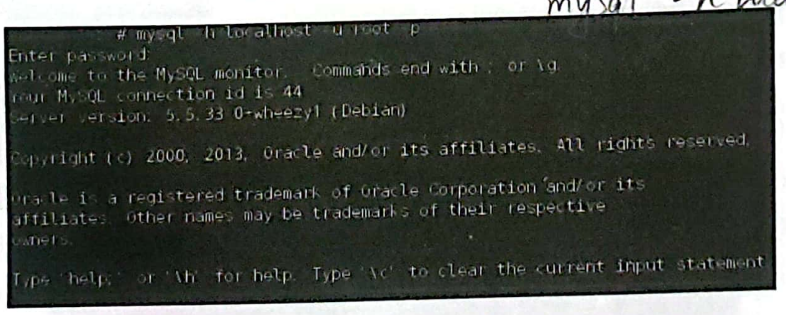

图5-47　连接数据库

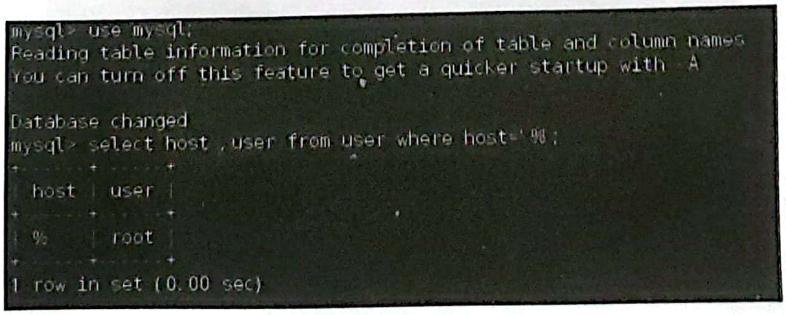

图5-48　进入系统MySQL数据库

pc-1

扫描全能王 创建

grant all on *.* to 'root'@'127.0.0.1';

```
mysql> revoke all on *.* from 'root'@'%' identified by 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all on *.* to 'root'@'127.0.0.1';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

图5-49　修改用户权限

（8）检查目标主机中是否存在数据库匿名用户，如果存在，则删除该用户。发现的数据库匿名用户信息及删除过程如图5-50所示。

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select Host,user from user where user='';
+------+------+
| Host | user |
+------+------+
| repo |      |
+------+------+
1 row in set (0.00 sec)

mysql> delete from user where user='';
Query OK, 1 row affected (0.00 sec)
```

图5-50　查找匿名用户

（9）禁止MySQL对本地文件进行存取，对mysqld的启动项进行加固。首先打开配置文件，如图5-51所示，修改对应参数，如图5-52所示。

```
[root@localhost ~]# find / -name my.cnf
/etc/mysql/my.cnf
[root@localhost ~]# vi /etc/mysql/my.cnf
[root@localhost ~]#
```

图5-51　打开MySQL配置文件

（10）限制一般用户浏览其他用户的数据库，对mysqld的启动项进行加固，操作如图5-53所示。

```
# This was formally known as [safe_mysqld]. Both versions are currently par
[mysqld_safe]
socket          = /var/run/mysqld/mysqld.sock
nice            = 0

[mysqld]
local_infile=0
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
-- 插入 --                                              32,15
```

图5-52　修改对应参数

扫描全能王 创建

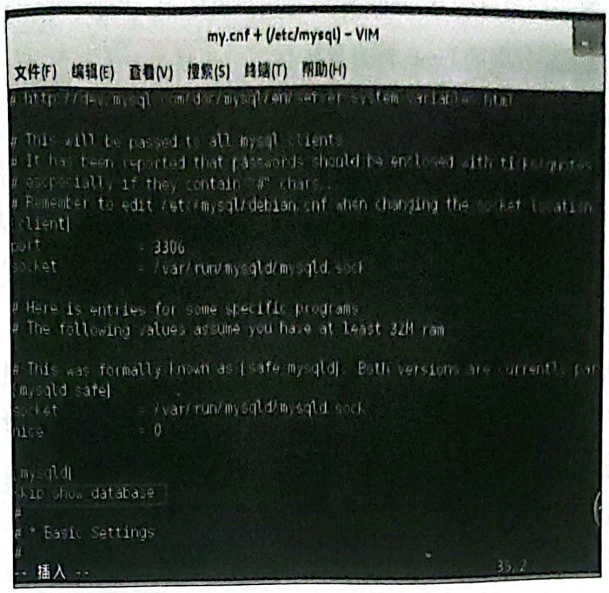

图5-53 修改限制一般用户浏览数据库权限

(11) MySQL密码管理。

密码是数据库安全管理的一个很重要因素，不要将纯文本密码保存到数据库中。因为如果你的计算机有安全危险，那么入侵者可以获得所有的密码并使用它们。相反，应使用MD5()、SHA1()或单向哈希函数。也不要从词典中选择密码，有专门的程序可以破解它们，应选用至少8位由字母、数字和符号组成的强密码。在存取密码时，使用MySQL的内置函数password()的sql语句对密码进行加密后存储。例如，以下列方式在users表中加入新用户：

```
#mysql > insert into users values (1,password(1234),'test');
```

(12) 使用独立用户运行MySQL。

绝对不要作为root用户运行MySQL服务器，这样做非常危险，因为任何具有FILE权限的用户都能够用root创建文件（例如，~root/.bashrc）。mysqld拒绝使用root运行，除非使用--user=root选项来明确指定。应该用普通非特权用户运行mysqld，为数据库建立独立的Linux中的MySQL账户，该账户只用于管理和运行MySQL。要想用其他Linux用户启动mysqld，则需要增加user选项指定/etc/my.cnf选项文件或服务器数据目录的my.cnf选项文件中的[mysqld]组的用户名。具体如下：

```
#vi /etc/my.cnf
[mysqld]
user=mysql
```

该命令使服务器用指定的用户来启动，无论是手动启动或通过mysqld_safe或mysql.server启动，都能确保使用mysql的身份。也可以在启动数据库时，加上user参数，具体命令为：

```
# /usr/local/mysql/bin/mysqld_safe -user=mysql &
```

如果是作为其他Linux用户（不用root运行mysqld），则不需要更改user表中的root用

扫描全能王 创建

户名，因为MySQL账户的用户名与Linux账户的用户名无关。确保mysqld运行时，只使用对数据库目录具有读或写权限的Linux用户来运行。

（13）禁止远程连接数据库。

在命令行netstat－ant下可以看到，默认的3306端口是打开的。此时，打开mysqld的网络监听，允许用户远程通过账号密码连接本地数据库（默认情况是允许远程连接数据库的）。为了禁止该功能，需启动skip－networking，不监听sql的任何TCP/IP的连接，切断远程访问的权利，以保证数据库的安全性。

假如需要远程管理数据库，则可以通过安装PhpMyadmin来实现。假如确实需要远程连接数据库，则至少需要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。命令如下：

```
vi /etc/my.cf 将#skip-networking 注释去掉。
# /usr/local/mysql/bin/mysqladmin -u root -p shutdown //停止数据库
#/usr/local/mysql/bin/mysqld_safe -user=mysql & //后台用mysql用户启动mysql
```

（14）用户目录权限限制。

默认的MySQL是安装在“/usr/local/mysql”目录下的，而对应的数据库文件在“/usr/local/mysql/var”目录下，因此，必须保证该目录不能让未经授权的用户访问后把数据库打包拷贝走。所以要限制对该目录的访问，以确保mysqld运行时，只使用对数据库目录具有读或写权限的Linux用户来运行。命令如下：

```
# chown -R root /usr/local/mysql/ //mysql主目录给root
# chown -R mysql.mysql /usr/local/mysql/var //确保数据库目录权限所属mysql用户
```

（15）限制连接用户的数量。

数据库的某用户多次远程连接，会导致性能的下降和影响其他用户的操作，有必要对其进行限制。可以通过限制单个账户允许的连接数量来实现，即通过设置my.cnf文件的mysqld中的max_user_connections变量来完成。GRANT语句也可以支持资源控制选项来限制服务器对一个账户允许的使用范围。

```
#vi /etc/my.cnf
[mysqld]
max_user_connections 2
```

（16）命令历史记录保护。

数据库相关的shell操作命令都会分别记录在.bash_history文件中，如果这些文件不慎被读取，则会导致数据库密码和数据库结构等信息泄露；而登录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，那么其也会被读取密码，因此需要删除这两个文件。同时，在进行登录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，建议将以上文件置空。命令如下：

```
# 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
```


扫描全能王 创建

(17) 禁止 MySQL 对本地文件存取。

在 MySQL 中，提供对本地文件的读取功能，使用的是“load data local infile”命令，在 5.0 版本中，该选项是默认打开的。该操作命令会利用 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。如果不需要读取本地文件，务必将其关闭。应该禁止在 MySQL 中使用“load data local infile”命令。网络上流传的一些攻击方法中就有用该命令的，同时它也是很多新发现的 SQL Injection 攻击利用的手段。黑客还能通过使用“load data local infile”命令装载“/etc/passwd”进一个数据库表，然后能用 SELECT 显示它，这个操作对服务器的安全来说是致命的。可以在 my. cnf 中添加 local - infile = 0，或者加参数“local - infile = 0”。命令如下：

```
#/usr/local/mysql/bin/mysqld_safe - user = mysql - local - infile = 0 &
#mysql > load data local infile 'sqlfile.txt' into table users fields terminated by ', ';
#ERROR 1148 (42000): The used command is not allowed with this MySQL version
```

- local - infile = 0 选项启动 mysqld 从服务器端禁用所有“load data local”命令，如果是获取本地文件，则可以打开，但是一般建议关闭。

(18) MySQL 服务器权限控制。

MySQL 权限系统的主要功能是证实连接到一台给定主机的用户，并且赋予该用户在数据库上的 SELECT、INSERT、UPDATE 和 DELETE 等权限。它的附加功能包括对 MySQL 特定的功能（例如 load data infile）进行授权及管理操作的能力。

管理员可以通过对 user、db、host 等表进行配置，来控制用户的访问权限，而 user 表权限是超级用户权限。只把 user 表的权限授予超级用户（如服务器或数据库主管）是明智的。对其他用户，应该把在 user 表中的权限设成“N”，并且仅在特定数据库的基础上授权。可以为特定的数据库、表或列授权，FILE 权限可以使用“load data infile”和“select…into outfile”语句读和写服务器上的文件，任何被授予 FILE 权限的用户都能读或写 MySQL 服务器能读或写的文件（说明：用户可以读任何数据库目录下的文件，因为服务器可以访问这些文件）。FILE 权限允许用户在 MySQL 服务器具有写权限的目录下创建新文件，但不能覆盖已有文件在 user 表的 File_priv 设置 Y 或 N。所以，当不需要读取服务器文件时，应关闭该权限。命令如下：

```
#mysql > load data infile 'sqlfile.txt' into table loadfile.users fields terminated by ',';
Query OK,4 rows affected (0.00 sec) //读取本地信息 sqlfile.txt'
Records: 4 Deleted: 0 Skipped: 0 Warnings: 0
#mysql > update user set File_priv = 'N' where user = 'root'; //禁止读取权限
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql > flush privileges; //刷新授权表
Query OK, 0 rows affected (0.00 sec)
#mysql > load data infile 'sqlfile.txt' into table users fields terminated by ', ';
//重登录读取文件
#ERROR 1045 (28000): Access denied for user 'root'@ 'localhost' (using password:
```


扫描全能王 创建

```
YES) //失败
#mysql >select * from loadfile.users into outfile 'test.txt' fields terminated by ', ';
ERROR 1045 (28000): Access denied for user 'root'@ 'localhost' (using password: YES)
```

为安全起见，随时使用 SHOW GRANTS 语句检查谁已经访问了什么，然后使用 REVOKE 语句删除不再需要的权限。

【相关知识】

1. MySQL 服务的启动和停止

具体命令如下：

启动：NET STOP MYSQL。

停止：NET START MYSQL。

2. 登录 MySQL

语法：mysql -u 用户名；-p 用户密码。

键入命令“mysql -u root -p”，按 Enter 键后会提示输入密码，输入“12345”，然后按 Enter 键即可进入 MySQL 中。mysql 的提示符是：mysql > 。

注意：如果是连接到另外的机器上，则需要加入参数“ -h 机器 IP”。

3. 增加新用户

增加新用户时的格式为：grant 权限 on 数据库 . * to 用户名@ 登录主机 identified by "密码"。例如，增加一个用户 user1，密码为 password1，让其可以在本机上登录，并对所有数据库有查询、插入、修改、删除的权限。首先以 root 用户连入 MySQL，然后键入以下命令：

```
grant select,insert,update,delete on *.* to user1@ localhost Identified by "password1"
```

如果希望该用户能够在任何机器上登录 MySQL，则将 localhost 改为%。如果不想 user1 有密码，则可以再打一个命令将密码去掉，命令如下：

```
grant select,insert,update,delete on mydb. * to user1@ localhost identified by ""
```

4. MySQL 中修改 root 密码的方法

（1）用 SET PASSWORD 命令进行修改，命令格式如下：

```
mysql -u root
mysql >SET PASSWORD FOR 'root'@ 'localhost' = PASSWORD ('newpass');
```

（2）用 mysqladmin 进行修改，命令格式如下：

```
mysqladmin -u root password " newpass"
```

如果 root 已经设置过密码，则采用如下方法：

```
mysqladmin -u root password oldpass " newpass"
```

（3）用 UPDATE 直接编辑 user 表，命令格式如下：

```
mysql -u root
mysql >use mysql;
mysql >UPDATE user SET Password = PASSWORD ('newpass') WHERE user ='root';
mysql >FLUSH PRIVILEGES;
```


扫描全能王 创建

(4) 在丢失 root 密码时，采用如下方法进行修改：

```
mysqld_safe --skip-grant-tables&
mysql -u root mysql
mysql >UPDATE user SET password=PASSWORD (" new password") WHERE user='root';
mysql >FLUSH PRIVILEGES;
```

5. 操作数据库

首先，登录到 MySQL 中，然后在 MySQL 的提示符下运行下列命令，每个命令以分号结束。

(1) 显示数据库列表的命令格式为：show databases;

缺省有两个数据库：MySQL 和 test。MySQL 库存放着 MySQL 的系统和用户权限信息，修改密码和新增用户实际上就是对这个库进行操作。

(2) 显示库中的数据表，命令格式为：

```
use mysql;
show tables;
```

(3) 显示数据表的结构，命令格式为：

```
describe 表名;
```

(4) 建库与删库，命令格式为：

```
create database 库名;
drop database 库名;
```

(5) 建表，命令格式为：

```
use 库名;
create table 表名 (字段列表);
drop table 表名;
```

(6) 清空表中记录，命令格式为：

```
delete from 表名;
```

(7) 显示表中的记录，命令格式为：

```
select * from 表名;
```

6. 导出和导入数据

(1) 导出数据，命令格式为：

```
mysqldump --opt test >mysql.test
```

意思为：将 test 数据库导出到 mysql.test 文件。后者是一个文本文件，如 mysqldump -u root -p123456 --databases dbname >mysql.dbname 就是把数据库 dbname 导出到文件 mysql.dbname 中。

(2) 导入数据，命令格式为：

```
mysqlimport -u root -p123456 <mysql.dbname
```

(3) 将文本数据导入数据库。

文本数据的字段数据之间用 Tab 键隔开，命令格式为：

```
use test;
load data local infile " 文件名" into table 表名;
```


扫描全能王 创建

7. mysqld 安全相关启动选项

（1）－local－infile［＝｛0｜1｝］：如果用－local－infile＝0 启动服务器，则客户端不能使用 local in load data 语句。

（2）－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，这个命令很有用。

（3）（OBSOLETE）－safe－show－database：在以前版本的 MySQL 中，该选项使 show databases 语句只显示用户具有部分权限的数据库名。在 MySQL 5.1 中，该选项不再作为现在的默认行为使用，有一个 SHOW DATABASES 权限可以用来控制每个账户对数据库名的访问。

（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。如果想让用户具有授权权限来创建新用户，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，而是必须使用 GRANT 语句给其他用户授予该权限。

（5）－secure－auth：不允许鉴定有旧（pre－4.1）密码的账户。

项目实训　电子商务网站 SQL 注入与防范

【任务描述】

电子商务网站一直是黑客入侵的主要目标，因为电子商务网站有用户购物消费信息，一旦网站被入侵成功，黑客就可以轻松获取用户的账号和登录密码，后续还可以获取用户付款信息等重要信息。因此电子商务网站加固与入侵防范非常重要。

【任务分析】

本任务通过对某电子商务网站进行漏洞入侵，学习掌握如何通过 SQL 手工注入来获取用户账号和密码。黑客入侵过程是先查找注入点，构造 SQL 语句获取网站管理员账号和密码，然后利用管理员账号和密码登录网站后台管理页面，对数据库进行操作，窃取用户数据，远程在网站服务器上创建管理员账户。管理员可以通过对 SQL 语句过滤来防范 SQL 注入攻击。

【任务实施】

（1）入侵者访问某目标电子商务网站，如图 5－54 所示。

（2）利用明小子等工具查找网站注入点。输入网址 http://www.ec.com/shop/goods.php?id＝9'，返回错误信息，从返回的错误信息可以看到后台使用的是 MySQL 数据库，并暴露了当前查询数据表及相关字段信息，如图 5－55 所示。

（3）构造 union 查询，登录 http://www.ec.com/shop/goods.php?id＝9' union select 1,2,3,4,5,6,7,8,9,10，之所以构造 10 个字段的 union 查询，是因为从上面的报错信息可以知道目标数据表查询出的字段为 10 个，如图 5－56 所示。

扫描全能王 创建