

## 20200623信息安全实训

笔记本： 我的第一个笔记本

创建时间： 2020/6/23 8:44

更新时间： 2020/6/23 17:59

作者： 820410740@qq.com

---

### mysql与php 测验

0.信息收集与扫描探测 (nmap,xray扫描探测工具)

1.sql注入 (原理, 注入类型, 基于报错的注入, 盲注, 手工注入)

sqlmap

(测验在线靶场试验) 生成报告 (步骤, 图片, 文字)

2.文件上传 (原理, 上传的类型)

3.抓包分析工具 (wireshark,科来) burpsuite

4.暴力破解 (原理, 工具)

5.系统提权

微软 access sqlserver

甲骨文 oracle mysql

mariadb 免费社区多人维护 kali linux 关系型数据库 解决我们应用系统内的复杂关系

redis mongodb 非关系型数据库

### Mysql与MariaDB

1. mysql之父 Michael Widenius

2. MariaDB是mysql的一个分支版本

3. Mysql商业授权, MariaDB免费开源。MariaDB与Mysql完全兼容, MariaDB在更新, 增加新特性与bug修复上要比Mysql更快速及时

## Mysql配置项

- 1.配置环境变量

- (1) MySQL 的bin路径放到环境变量当中

- (2) mysql -v输出版本号, 表示配置成功

- 2.MySQL 配置

- (1) skip-grant-tables 数据库启动的时候 跳跃权限表的限制, 不用验证密码, 直接登录。这种情况只有在忘记root密码 不得已重启数据库的情况下使用的。现网环境慎用, 需要重启数据库, 并且安全性低。

- (2) skip-networking 关闭MySQL的TCP/IP连接方式, 开启该选项后就不能远程访问MySQL

- (3) bind-address 指定的IP访问, 前提是关闭skip-networking

- (4) log-error 记载着服务器启动和关闭的情况, 还记载着关于服务器故障或异常状况信息, 如果服务器无法启动, 首先应该查看该日志。在数据库意外发生时, 服务器会在结束之前把一条信息写入错误日志以表明发生了什么问题, 错误日志路径, 不需要文件后缀名。

- (5) general\_log, general\_log\_file 将所有到达MySQL Server的SQL语句记录下来，一般不会开启该功能，因为log的量会非常庞大。但个别情况下可能会临时的开一会儿general\_log以供排障使用。
- (6) slow\_query\_log, long\_query\_time, slow\_query\_log\_file 慢查询日志 slow\_query\_log开启，根据long\_query\_time配置项设置sql语句查询时间，超过设置时间将sql语句记录在slow\_query\_log\_file设置的文件日志里。

重置密码的时候 skip-grant-tables

service mysql start/stop/restart 开启mysql服务  
 /etc/init.d/mysql start/stop/restart 开启mysql服务  
 /etc/init.d/mysql status 查看状态

mysql\_secure\_installation 初始化设置密码

## MYSQL基础命令

mysql连接 (mysql -u用户名 -p密码 -default-character-set=utf8)

显示所有配置项 (show global variables)

显示数据库 (show databases)

创建数据库 (create database 数据库名)

删除数据库 (drop database 数据库名)

使用数据库 (use 数据库名)

查看表 (show tables)

显示常用信息 (select user(),version(),database())

创建表 (create table <表名> ( <字段名1> <类型1> [...<字段名n> <类型n>]))

删除表 (drop table <表名>)

显示表结构 (describe 表名)

# MYSQL 常用数据类型

- 1.TINYINT          1 字节      (-128, 127)      (0, 255)
- 2.SMALLINT        2 字节      (-32 768, 32 767)      (0, 65 535)
- 3.MEDIUMINT      3 字节      (-8 388 608, 8 388 607)      (0, 16 777 215)
- 4.INT或INTEGER 4 字节      (-2 147 483 648, 2 147 483 647)      (0, 4 294 967 295)
- 5.CHAR            0-255字节      定长字符串
- 6.VARCHAR        0-65535 字节      变长字符串
- 7.BLOB            0-65535字节      二进制形式的长文本数据
- 8.TEXT            0-65535字节      长文本数据

## 导入/导出数据备份命令

导入数据库 (source 完整脚本路径)

导出数据库 (mysqldump -u用户名 -p密码 -d不带数据 -add-drop-table 数据库名 > sql脚本名称)

## MYSQL 权限控制


- 数据库mysql, 主要授权表user, host, db, tables\_priv, columns\_priv.
- create user 用户名@主机名 IDENTIFIED BY 密码;
- grant 权限1,权限2... on 数据库.数据表 to 用户名@主机 IDENTIFIED BY 密码;
- with grant option; 是否可授权他人权限。
- flush privileges; 刷新权限。
- revoke 权限 on 数据库.数据表 from 用户名@主机; 回收权限。
- drop user 用户名@主机; 删除用户。

## SQL语句之MYSQL CURD操作

- select 字段1,字段2... from 表名 where < 表达式 > 查询
- update 表名 set 字段=新值,... where < 表达式 > 修改
- delete from 表名 where < 表达式 > 删除
- insert into 表名(字段1,字段2...) values (字段值1,字段值2...); 插入

# SQL语句

- 1. WHERE 子句 有条件地从表中选取数据，可将 WHERE 子句添加到 SELECT 语句中。
  - `SELECT field1, field2,...fieldN FROM table_name1, table_name2...`
  - `[WHERE condition1 [AND [OR]] condition2.....`
  - 注：WHERE 子句也可以运用于 SQL 的 DELETE 或者 UPDATE 命令。
- 2. UNION 操作符 用于连接两个以上的 SELECT 语句的结果组合到一个结果集合中。多个 SELECT 语句会删除重复的数据。
  - `SELECT field1, field2, ... fieldN FROM tables1 [WHERE conditions] UNION`
  - `[ALL | DISTINCT]`
  - `SELECT field1, field2, ... fieldN FROM tables2 [WHERE conditions];`
- 3. UNION ALL 操作符
  - UNION 语句：用于将不同表中相同列中查询的数据展示出来；（不包括重复数据）
  - UNION ALL 语句：用于将不同表中相同列中查询的数据展示出来；（包括重复数据）

- 
- 4. ORDER BY 对读取的数据进行排序。
    - `SELECT field1, field2,...fieldN FROM table_name1, table_name2...`
    - `ORDER BY field1 [ASC [DESC]][默认 ASC]], [field2...] [ASC [DESC]][默认 ASC]]`
  - 5. GROUP BY 对结果集进行分组。
    - `SELECT column_name, function(column_name) FROM table_name`
    - `WHERE column_name operator value GROUP BY column_name;`
  - 6. WITH ROLLUP 以实现在分组统计数据基础上再进行相同的统计（SUM,AVG,COUNT...）。

## MYSQL 字符串函数

- 1. CONCAT(s1,s2...sn) 合并多个字符串。
  - `SELECT CONCAT("hello", 0x7C, "mysql") AS SHOWSTRING;`
- 2. CONCAT\_WS(x, s1,s2...sn) 合并多个字符串，并添加分隔符。
  - `SELECT CONCAT_WS('_', "hello", 0x7C, "mysql") AS SHOWSTRING;`
- 3. ASCII(s) 返回字符串 s 的第一个字符的 ASCII 码。
  - `SELECT ASCII('Hello,Mysql') AS SHOWSTRING;`
- 4. HEX(x) 返回x的16进制编码



# MYSQL 元数据

- information schema数据库是MySQL自带的，它提供了访问数据库元数据的方式。元数据是关于数据的数据，如数据库名或表名，列的数据类型，或访问权限等。有些时候用于表述该信息的其他术语包括“数据词典”和“系统目录”。
- 在MySQL中，把 information schema 看作是一个数据库，确切说是信息数据库。其中保存着关于MySQL服务器所维护的所有其他数据库的信息。如数据库名，数据库的表，表栏的数据类型与访问权限等。在INFORMATION SCHEMA中，有数个只读表。它们实际上是视图，而不是基本表，因此，你将无法看到与之相关的任何文件。
- information\_schema数据库主要表说明：
  - SCHEMATA表：提供了当前mysql实例中所有数据库的信息。
  - TABLES表：提供了关于数据库中的表的信息（包括视图）。详细表述了某个表属于哪个schema，表类型，表引擎，创建时间等信息。
  - COLUMNS表：提供了表中的列信息。详细表述了某张表的所有列以及每个列的信息。
  - USER PRIVILEGES（用户权限）表：给出了关于全程权限的信息。该信息源自mysql.user授权表。是非标准表。