

## 第二章 原理篇

### 第一节：sql注入基础

#### 1.1 前言

从本节开始，讲开始第二章web漏洞原理篇的讲解。首先带给大家的是sql注入漏洞。sql注入漏洞是web层面最高危的漏洞之一。2008年至2018年期间，sql注入漏洞连续三年位于owasp漏洞排行榜中的第一名。

#### 1.2 免责声明

该课程中涉及的技术只适合于CTF比赛和有合法授权的渗透测试。请勿用于其他非法用途，如果作于其他非法用途，与本文作者无关。

#### 1.3什么是sql注入

在上节的课程中，已经带领大家学习了基本的sql语句使用。sql注入简单的来说，就是普通用户在web界面可以通过某些操作来执行用户输入的恶意的sql语句以达到攻击的目的。

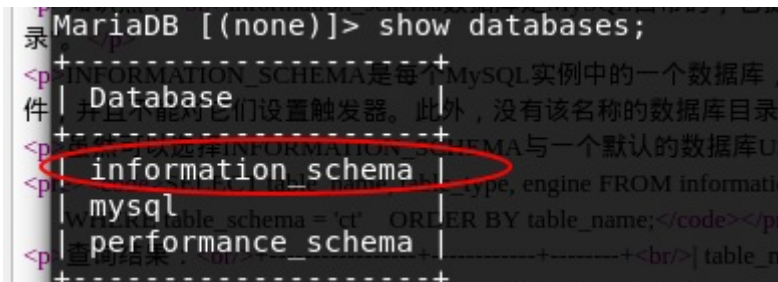
#### 1.4 INFORMATION\_SCHEMA

mysql数据库注入大概是以5.0版本为分界线。5.0以下的版本是没有INFORMATION\_SCHEMA这个系统表，只能通过暴力去跑表名，而5.0以上的版本可以通过查询INFORMATION\_SCHEMA这个表来获取数据库的相关信息。关于mysql数据库，在关于mysql的注入中，发现一个注入点以后，大多数的操作都是查询INFORMATION\_SCHEMA这个表中的相关信息。

##### 1.4.1 简介

INFORMATION\_SCHEMA是MYSQL数据库中自带的，它提供了访问数据库元数据数据的方式。元数据库即数据的数据，如数据库的库名或者表名、列的数据类型或者访问权限等。

INFORMATION\_SCHEMA是每个MySQL实例中的一个数据库，它存储有关MySQL服务器维护的所有其他数据库的信息。该INFORMATION\_SCHEMA数据库包含几个只读表。它们实际上是视图，而不是基表，所以没有与它们相关联的文件。但是对于INFORMATION\_SCHEMA，只能读取表的内容，不执行INSERT，UPDATE或DELETE对他们的操作。



```

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
```

##### 1.4.2 常用表

```
MariaDB [information_schema]> show tables;
```

Tables_in_information_schema
ALL_PLUGINS
APPLICABLE_ROLES
CHARACTER_SETS
CHECK_CONSTRAINTS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
COLUMN_PRIVILEGES
ENABLED_ROLES
ENGINES
EVENTS
FILES
GLOBAL_STATUS
GLOBAL_VARIABLES
KEY_CACHES
KEY_COLUMN_USAGE
PARAMETERS
PARTITIONS
PLUGINS
PROCESSLIST
PROFILING
REFERENTIAL_CONSTRAINTS
ROUTINES
SCHEMATA
SCHEMA_PRIVILEGES
SESSION_STATUS
SESSION_VARIABLES
STATISTICS
SYSTEM_VARIABLES
TABLES
TABLESPACES
TABLE_CONSTRAINTS
TABLE_PRIVILEGES
TRIGGERS
USER_PRIVILEGES
VIEWS
GEOMETRY_COLUMNS
SPATIAL_REF_SYS
CLIENT_STATISTICS
INDEX_STATISTICS
TINNODB_SYS_DATAFILES

SCHEMATA表：提供了当前mysql实例中所有数据库的信息。是show databases的结果取之此表。

```

INFORMATION_SCHEMA是MySQL数据库中自带的，它提供了访问数据库
mysql> select SCHEMA_NAME from schemata;
+-----+
| SCHEMA_NAME |
+-----+
| information_schema |
| challenges |
| dwwa |
| mysql |
| performance_schema |
| security |
| test |
+-----+
7 rows in set (0.00 sec)

```

INFORMATION\_SCHEMA表：提供了当前mysql实例中所有数据库的信息。是show databases的结果取之此表。

```

23 rows in set (0.002 sec)
MariaDB [information_schema]> select TABLE_NAME from tables;
+-----+
| TABLE_NAME |
+-----+
| ALL_PLUGINS |
| APPLICABLE_ROLES |
| CHARACTER_SETS |
| CHECK_CONSTRAINTS |
| COLLATIONS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS |
| COLUMN_PRIVILEGES |
| ENABLED_ROLES |
| ENGINES |
| EVENTS |
| FILES |
| GLOBAL_STATUS |
| GLOBAL_VARIABLES |
| KEY_CACHES |
| KEY_COLUMN_USAGE |
| PARAMETERS |
| PARTITIONS |
| PLUGINS |
| PROCESSLIST |
| PROFILING |
| REFERENTIAL_CONSTRAINTS |
| ROUTINES |

```

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。是show columns from schemaname.tablename的结果取之此表。

```

MariaDB [information_schema]> select COLUMN_NAME from COLUMNS;
+-----+
| COLUMN_NAME |
+-----+
| INFORMATION_SCHEMA |
| COLUMNS |
| COLUMNS_PRIVILEGES |
| COLUMNS_USAGE |
| CHARACTER_SETS |
| CHARACTER_SET_APPLICABILITY |
| CONSTRAINTS |
| KEY_COLUMN_USAGE |
| ROUTINES |
| TABLES |
| TABLES_PRIVILEGES |
| TRIGGERS |
| VIEWS |
+-----+

```

INFORMATION\_SCHEMA 是每个MySQL实例中的一个数据库，它存储有关MySQL服务器维护的所有其他数据库的信息。对它们设置触发器。此外，没有该名称的数据库目录。

虽然可以选择INFORMATION\_SCHEMA与一个默认的数据库USE语句，只能读取表的内容，不执行INSERT，UPDATE，DELETE。

```

14 | PLUGIN_NAME | table_name | table_type | engine |
14 | PLUGIN_VERSION | schema = 'ct' | ORDER BY table_name;
14 | PLUGIN_STATUS |
14 | PLUGIN_TYPE | MyISAM | user2 | BASE TABLE | MyISAM |
14 | PLUGIN_TYPE_VERSION |
14 | PLUGIN_LIBRARY |
14 | PLUGIN_LIBRARY_VERSION |
14 | PLUGIN_AUTHOR |
14 | PLUGIN_DESCRIPTION |
14 | PLUGIN_LICENSE |
14 | LOAD_OPTION |
14 | PLUGIN_MATURITY |
14 | PLUGIN_AUTH_VERSION |
14 | GRANTEE |
14 | ROLE_NAME |
14 | IS_GRANTABLE |
14 | IS_DEFAULT |
14 | CHARACTER_SET_NAME |
14 | DEFAULT_COLLATE_NAME |
14 | DESCRIPTION |
14 | MAXLEN |
14 | CONSTRAINT_CATALOG |
14 | CONSTRAINT_SCHEMA |
14 | CONSTRAINT_NAME |
14 | TABLE_NAME |
14 | CHECK_CLAUSE |
14 | COLLATION_NAME |
14 | CHARACTER_SET_NAME |
14 | ID |
14 | IS_DEFAULT |
14 | IS_COMPILED |
14 | SORTLEN |
14 | COLLATION_NAME |
14 | CHARACTER_SET_NAME |
14 | TABLE_CATALOG |
14 | TABLE_SCHEMA |
14 | TABLE_NAME |
14 | COLUMN_NAME |
14 | ORDINAL_POSITION |
14 | COLUMN_DEFAULT |
14 | IS_NULLABLE |

```

SELECT table\_name, table\_type, engine FROM information\_schema.tables

库表说明:

DATA表：提供了当前mysql实例中所有数据库的信息，show databases查询的结果取之此表。

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、表引擎版本、表引擎选项、表引擎成熟度、表引擎授权版本、表引擎授权人、表引擎授权表、表引擎授权默认、表引擎授权字符集、表引擎授权默认校对名、表引擎授权描述、表引擎授权最大长度、表引擎授权约束目录、表引擎授权约束模式、表引擎授权约束名称、表引擎授权表名称、表引擎授权检查子句、表引擎授权校对名称、表引擎授权字符集名称、表引擎授权ID、表引擎授权是否默认、表引擎授权是否编译、表引擎授权排序长度、表引擎授权校对名称、表引擎授权字符集名称、表引擎授权表目录、表引擎授权表模式、表引擎授权表名称、表引擎授权列名称、表引擎授权列序数、表引擎授权列默认值、表引擎授权是否可为空。

提供了关于表索引的信息。是show index from schemaname.tablename的结果取之此表。

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。

PRIVILEGES（方案权限）表：给出了关于方案（数据库）权限的信息。该信息来自mysql.db授权表。是非标准表。

PRIVILEGES（表权限）表：给出了关于表权限的信息。该信息源自mysql.tables\_priv授权表。是非标准表。

PRIVILEGES（列权限）表：给出了关于列权限的信息。该信息源自mysql.columns\_priv授权表。是非标准表。

CHARACTER SETS（字符集）表：提供了mysql实例可用字符集的信息。是SHOW CHARACTER SET结果集取之此表。

CHARACTER SET APPLICABILITY表：指明了可用于校验的字符集。这些列等效于SHOW COLLATION结果集取之此表。

CONSTRAINTS表：描述了存在约束的表。以及表的约束类型。

KEY\_COLUMN\_USAGE表：描述了具有约束的键列。

ROUTINES表：提供了关于存储子程序（存储程序和函数）的信息。此时，ROUTINES表不包含自定义函数（UDF）。

TABLES表：关于数据库中的视图的信息。需要有show view权限，否则无法查看视图信息。

TABLES\_PRIVILEGES表：提供了关于触发程序的信息。必须有super权限才能查看该表。

TABLE\_SCHEMA前数据库用户

TABLE\_NAMEqlmap.py -u "http://shan\*\*\*\*ui-china.com/jianzhangdetail.php?id=20" --current-user

COLUMN\_NAME示，获取当前数据库用户为"shan\*\*\*@#"。

ORDINAL\_POSITIONk,size 16,text QDUxQ1RP5Y2a5a6i,color BFFFFFF,t 100,g se,x 10,y 10,shadow 90,type ZmFu2

COLUMN\_DEFAULT数据库表

IS\_NULLABLEimap.py -u "http://shan\*\*\*\*-china.com/jianzhangdetail.php?id=20" -D shan\*\*\*\* --tables

这几个在日后sql注入的过程会经常用到，其他的不在此做详细的介绍，如果想要了解更多：  
<http://help.wopus.org/mysql-manage/607.html>

## 1.5 搭建一个sql注入的靶场

靶场推荐使用sqli-labs。靶场下载地址：<https://github.com/Audi-1/sqli-labs>

### 1.5.1 window平台搭建

1.推荐使用集成环境搭建，我将使用phpstudy进行安装。下载地址：<http://down.php.cn/PhpStudy20180211.zip>

2.下载以后打开压缩包,解压缩。然后点击phpStudySetup进行安装。这里我将其安装到了c盘下。

3.安装完成以后，运行软件。然后点击启动。



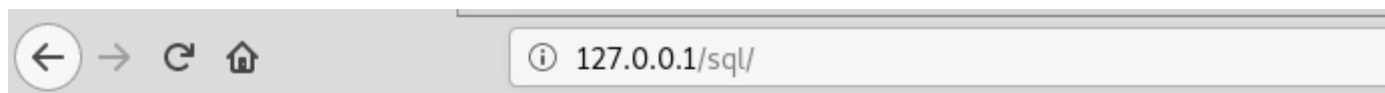
4.下载靶场文件

5.点击其他选项菜单，然后选择网站根目录,讲刚刚的下载的文件解压缩后放入里面





- 6.将解压缩后的文件重命名为sql
- 7.进入sql文件夹，找到名为sql-connections的文件夹。然后进入,找到文件db-creds.inc
- 7.修改"\$dbpass = ''" 为 "\$dbpass='root'"
- 8.保存文件，然后在浏览器输入：127.0.0.1/sql访问即可。
- 9.点击第二个。



## **SQLi-LABS Page-1(Basic Challenges)**

[Setup/reset Database for labs](#)

[Page-2 \(Advanced Injections\)](#)

[Page-3 \(Stacked Injections\)](#)

[Page-4 \(Challenges\)](#)

- 10.当出现如下字样说明安装成功。

## SETTING UP THE DATABASE SCHEMA AND POPULATING DATA IN TABLES:

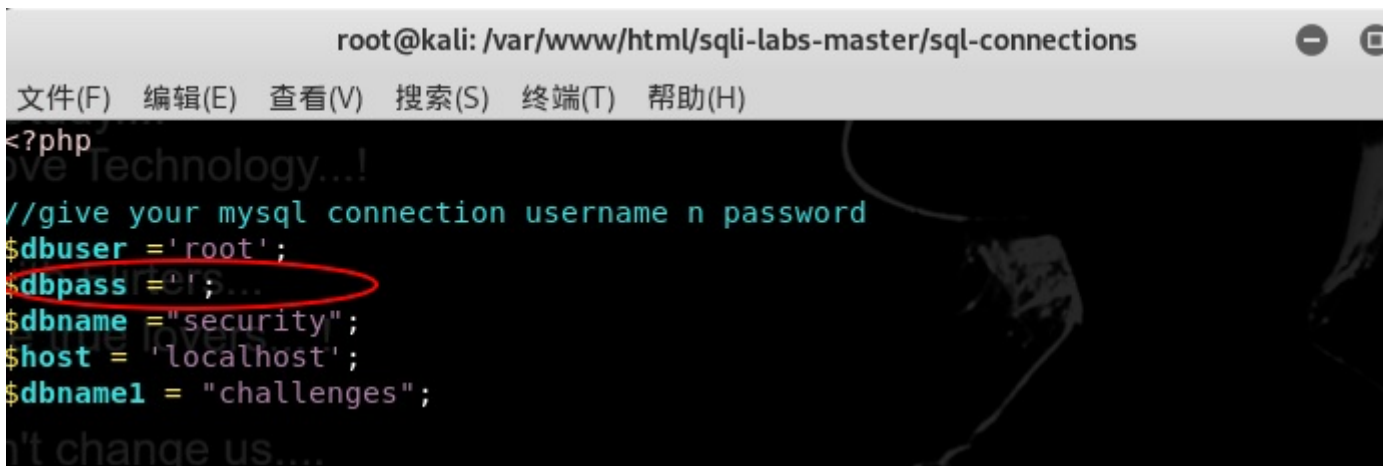
```
[*].....Old database 'SECURITY' purged if exists
[*].....Creating New database 'SECURITY' successfully
[*].....Creating New Table 'USERS' successfully
[*].....Creating New Table 'EMAILS' successfully
[*].....Creating New Table 'UAGENTS' successfully
[*].....Creating New Table 'REFERERS' successfully
[*].....Inserted data correctly into table 'USERS'
[*].....Inserted data correctly into table 'EMAILS'
[*].....Old database purged if exists
[*].....Creating New database successfully
[*].....Creating New Table '6Z5LRKRLFJ' successfully
[*].....Inserted data correctly into table '6Z5LRKRLFJ'
[*].....Inserted secret key 'secret_YW5E' into table
```

### 1.5.2 linux平台搭建

首先声明一点，linux 平台的mysql还有php版本不要太高。否则将会安装不成功。这里我将采用 apache2+mysql+php的环境，至于环境的安装再次不过多叙述。

- 1.cd /var/www/html/
- 2 .wget wget <https://codeload.github.com/Audi-1/sqli-labs/zip/master>
- 3.mv master master.zip
- 4.unzip master.zip
- 5.cd sqli-labs-master/
- 6.cd sql-connections/
- 7.vi db-creds.inc

8.将红圈处改为刚刚配置数据库时设置的密码。



```
root@kali: /var/www/html/sqli-labs-master/sql-connections
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
<?php
//give your mysql connection username n password
$dbuser = 'root';
$dbpass = '';
$dbname = "security";
$host = 'localhost';
$dbname1 = "challenges";
```

9.保存退出。

10.cd /var/www/html/

11.mv sqli-labs-master/ sql

12.然后在浏览器访问 :ip/sql

注意事项：

1.安装完成以后如果访问失败，可以尝试重启apache

service apache2 restart

2.重启apache还是不可以的话，可以尝试关闭系统防火墙做测试。如果可以的话，打开指定的端口即可。

3.不建议坐在公网服务器上，最好本地搭建。如果坐在公网服务器，建议坐在docker里面。

### 1.5.3 靶场地址

如果大家没有时间搭建靶场的话，可以使用玄魂工作室提供的靶场：<http://39.98.88.18:8080/sql/>

### 1.6 课后习题

本章节讲解的是基础知识，所以没有实战的ctf题目供大家玩耍。如果大家没有mysql数据库可以玩耍的话，可以使用我们上次课程那个库。题目地址是

<http://39.98.88.18/challenges#MYSQL%E5%9F%BA%E7%A1%80>

从下节课开始，我们讲开始讲解sql注入。在正式开始课程之前,希望大家能够思考几个问题，我将会在下节课程中给出答案。

1.sql注入的最终目的是什么

2.我们如何才能实现这个最终目的。