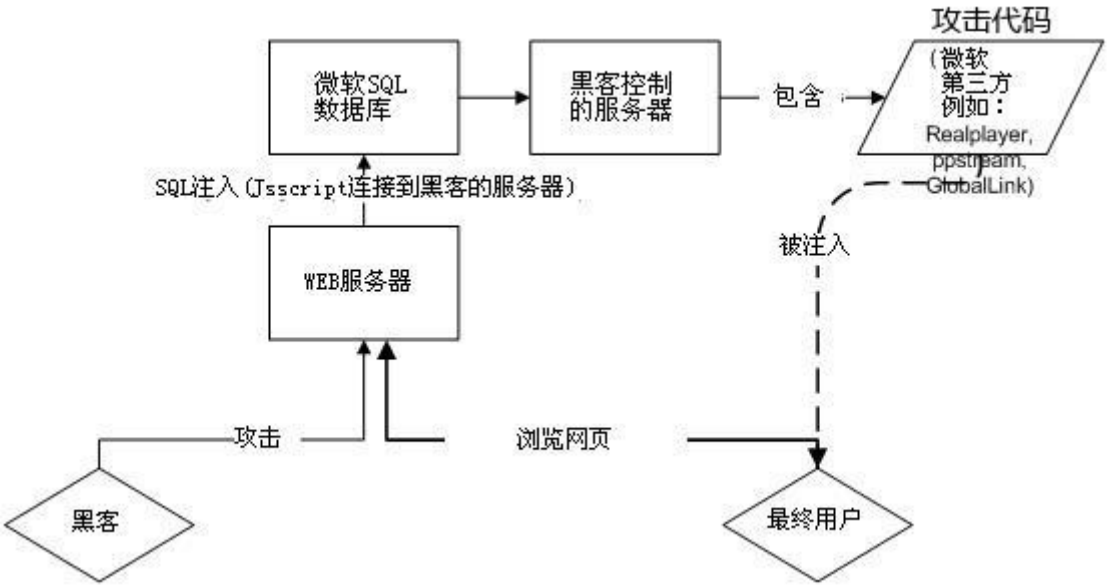
# SQL注入攻击

## 什么是SQL注入攻击?




> 攻击者把SQL命令插入到Web表单的域或页面请求的查询字符串中，欺骗服务器执行恶意的SQL命令。由于SQL注入攻击利用的是合法的SQL语句，是的这种攻击不能够被防火墙检查出来，而且由于对任何基于SQL语言标准的数据库都适用，所以危害特别大。

## 注入攻击类型

### 1.没有过滤转义字符

用户的输入没有进行转义字符过滤，而被传递给SQL语句，从而导致攻击者对数据库的操作，例如：

```
SELECT * FROM student WHERE name = ?
```

假如用户将name恶意伪造为

```
wanghao' or '123'='123
```

此时SQL语句变化为：

```
SELECT * FROM student WHERE name = 'wanghao' or '123'='123'
```

因为'123'='123'是正确的，如果这段代码被用语一个认证过程，那么就能强迫选择一个合法的用户名。

### 2.错误类型

没有强制约束类型，例如：

```
UPDATE course SET score = ? where name = 'wanghao'
```

如果输入的成绩被恶意伪造为：

```
80;DROP TABLE subject; #
```

此时SQL语句变化为：

```
UPDATE course SET score = 80;DROP TABLE subject; # where name = 'wanghao'
```

它会将'subject'表从数据库中删除

### 3.条件性差错

一种盲目的SQL注入，如果WHERE语句为真，SQL注入会迫使数据库执行一个引起错误的语句，从而导致一个SQL错误。例如：

```
SELECT 1/0 FROM course WHERE Cname='wanghao'
```

当用户wanghao存在时，被零除将会导致错误。

### 4.时间延误

一种盲目的SQL注入，根据所注入的逻辑，可以导致SQL引擎执行一个长队列或者是一个时间延误语句。

... ... ...

## 如何防范

- 根据内容构造SQL命令之前，对用户输入进行验证（利用正则表达式等）与替换。
  - 替换单引号，即把所有单独出现的单引号改成两个单引号，防止攻击者修改SQL命令的含义。
  - 删除用户输入内容中的所有连字符，防止攻击者顺利获得访问权限。
- 加密处理：对查询字符串、用户登录名称、密码等进行加密处理；
- 限制权限：对于用来执行查询的数据库账户，限制其权限。
- 合法检查：检查用户输入的合法性，确信输入的内容只包含合法的数据
- 数量记录：检查提取数据的查询所返回的记录数量。