# 第三次实验

## 实验一

### Task1

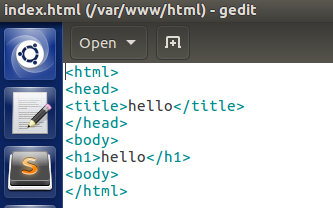
sudo apt-get install apache2

安装apache2，并选择默认配置

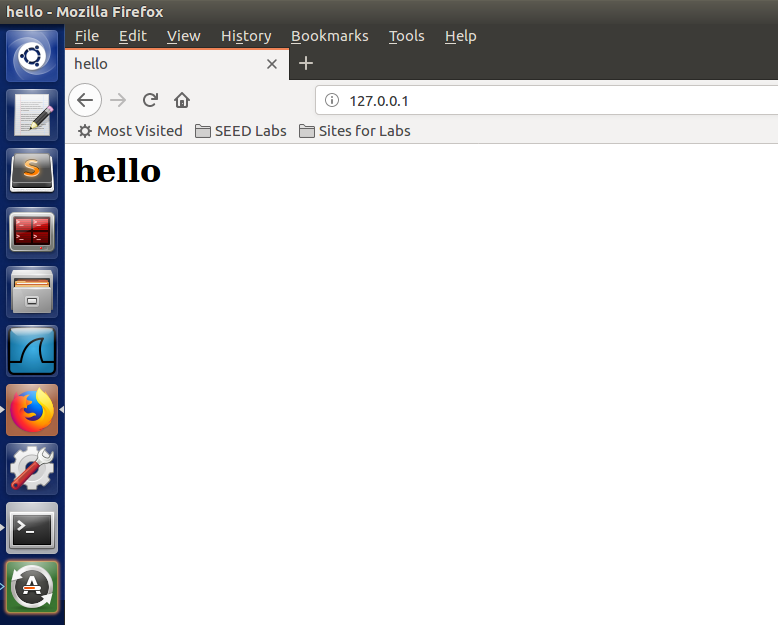
cd /var/www/html

sudo gedit index.html

修改apache默认根目录下的index.html

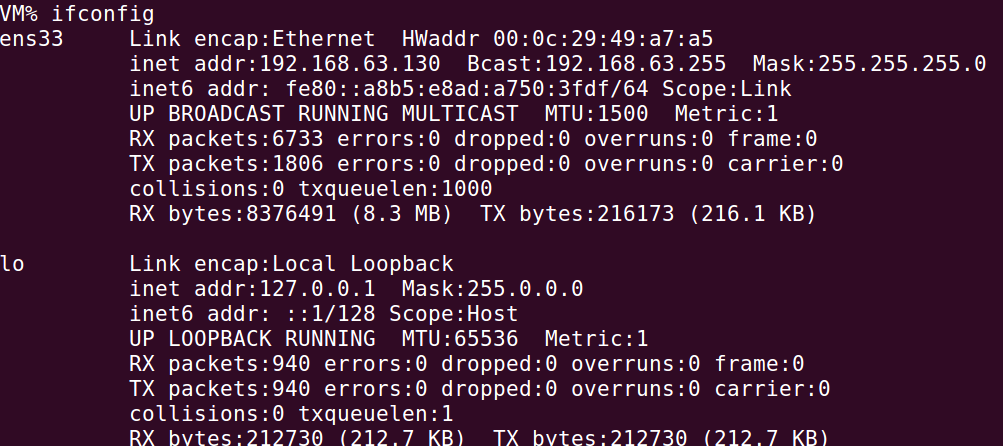


刷新127.0.0.1，获得我们修改的html页面.



### Task2

Ifconfig查看ip

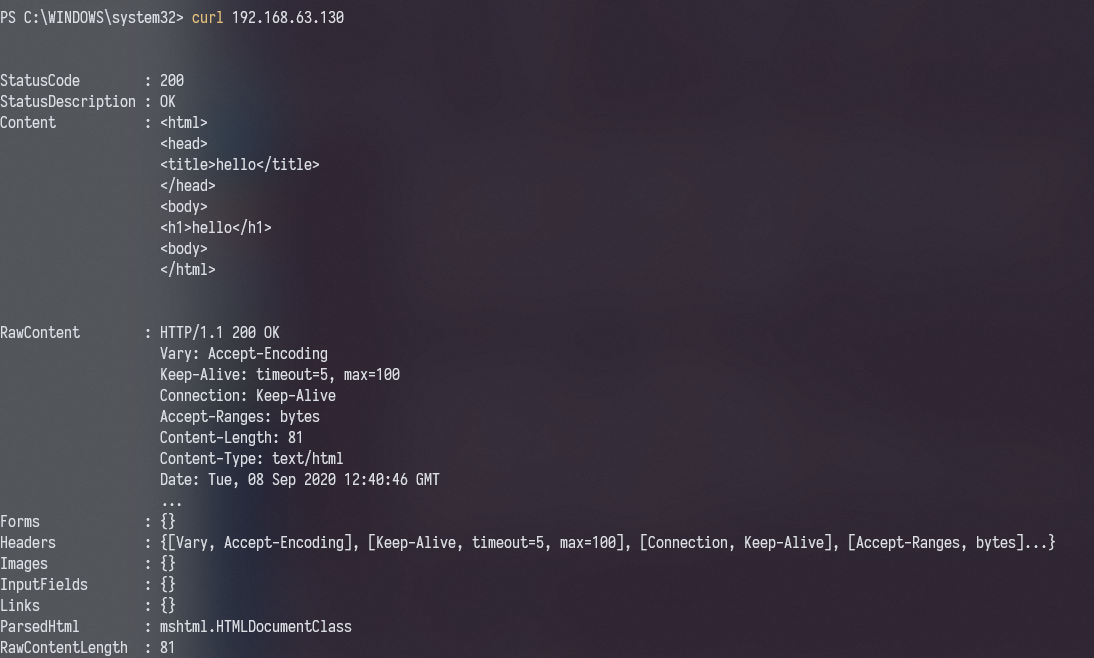


修改hosts文件



### Task3

Powershell中 curl 192.168.63.130 可以得到如下结果，可以看到我们访问html成功。



Python 3.5已经安装



安装依赖

pip3 install requests requests\_toolbelt

写入代码

import requests

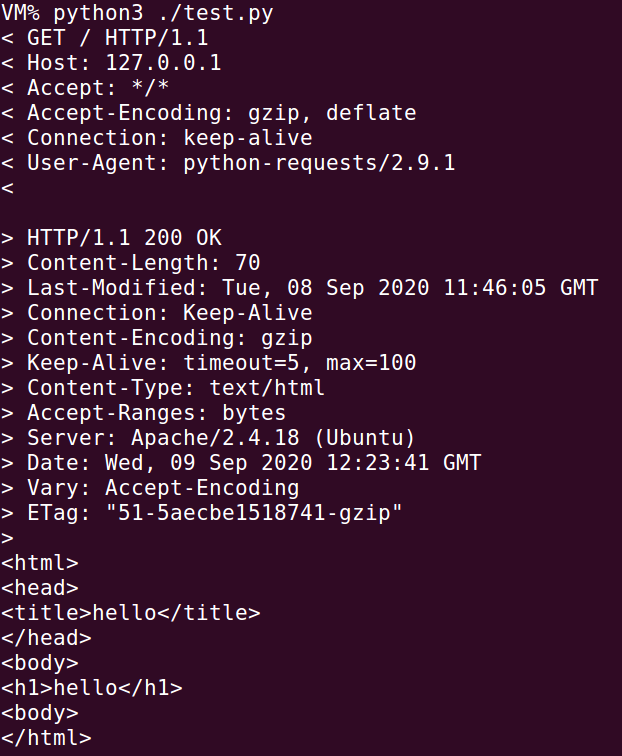
from requests\_toolbelt.utils import dump

resp =requests.get('http://127.0.0.1')

data =dump.dump\_all(resp)

print(data.decode('utf-8'))

执行脚本，输出如下，可以看到执行正确，获得了页面。



### Task4

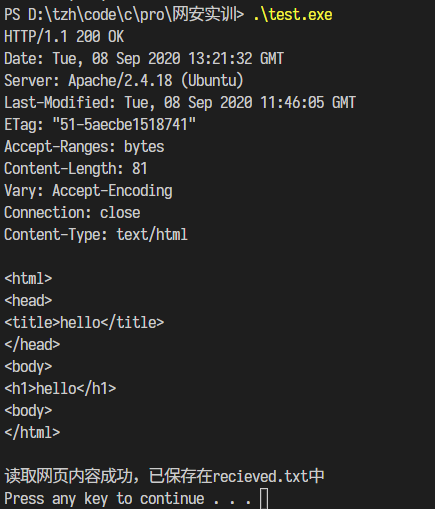
程序修改为cpp文件，手册有误。

修改void main到int main  然后修改char\*到const char\*，改完报错位置。

Windows下使用mingw中的g++进行编译，需要加上 -lwsock32选项。



运行程序，状态码200，成功。网页定向正确。



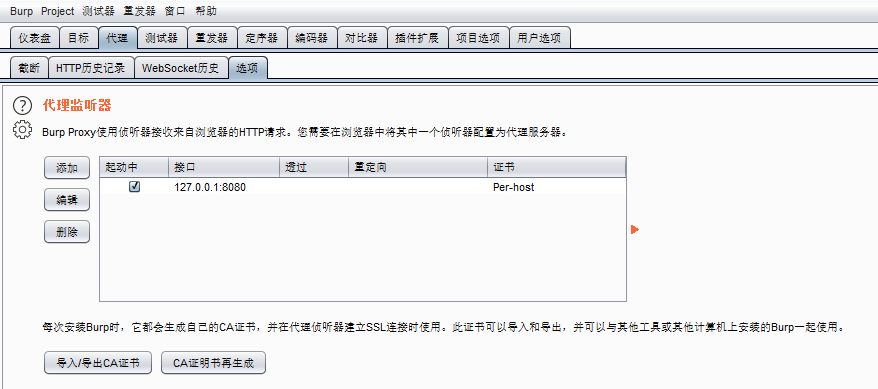
### Task5

安装java8环境。（需要是java8）

通过插件设置浏览器代理端口为8080端口。



Burp处设置代理8080，并导入证书。



浏览器设置，搜索证书，点击安全，管理证书，导入到受信任的根证书颁发机构。

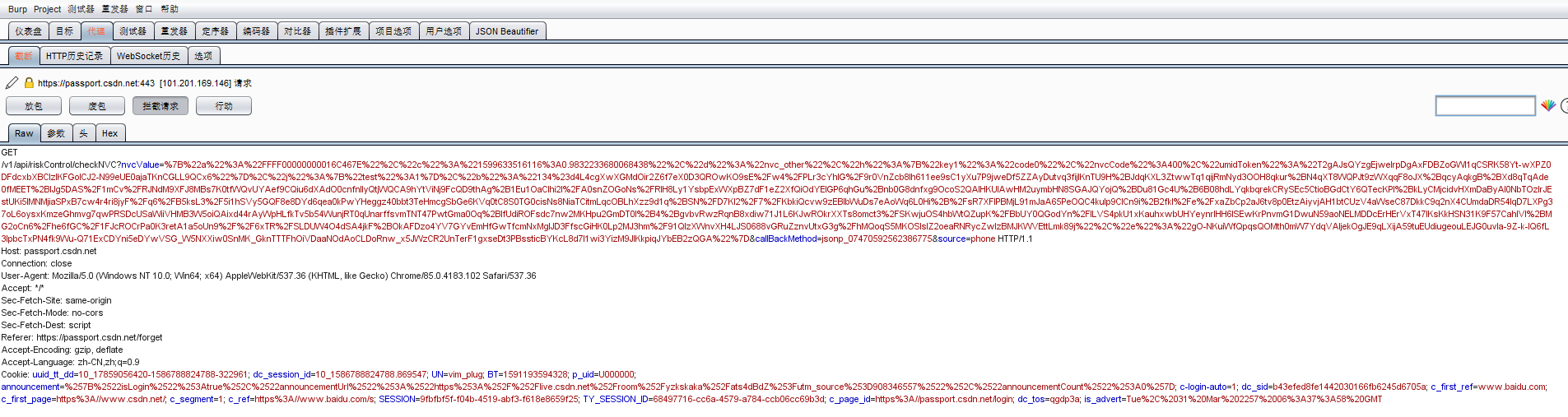


#### 访问 my.seu.edu.cn，请求被拦截如下。

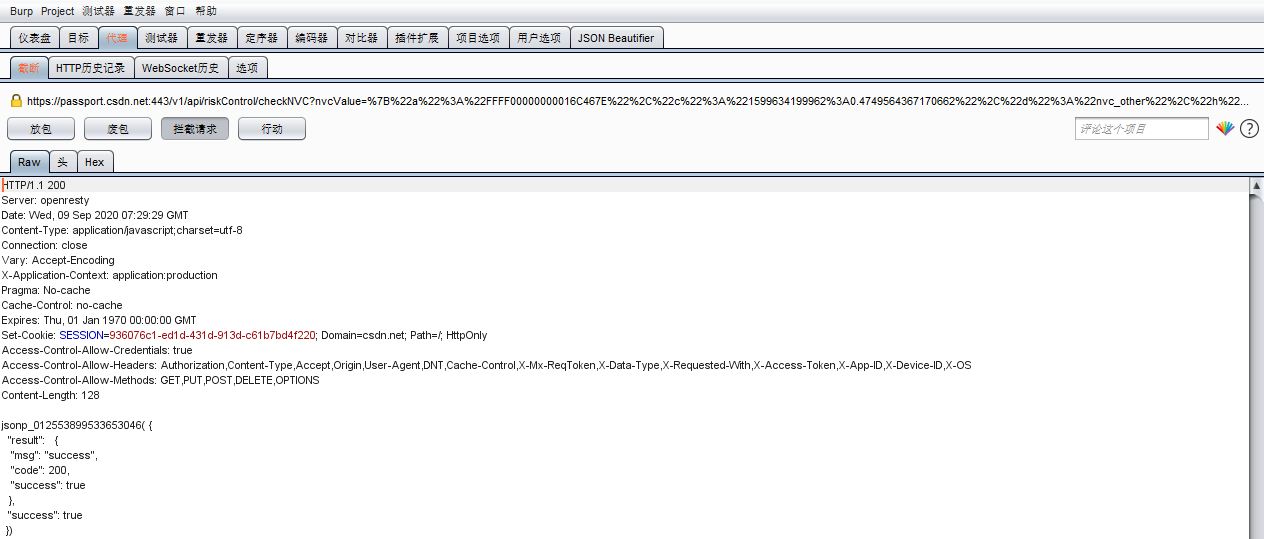


#### 测试csdn的找回密码功能。

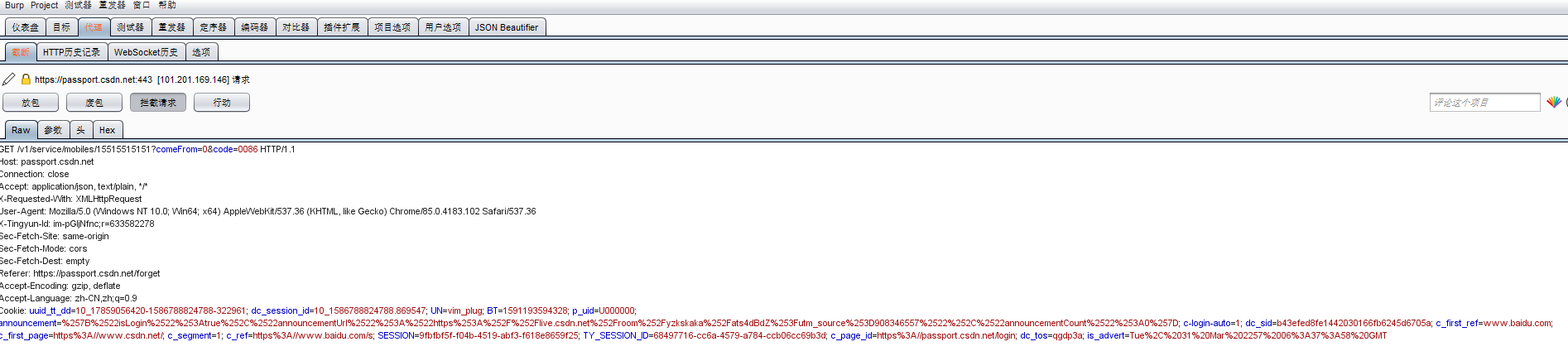
首先向一个api发送get报文



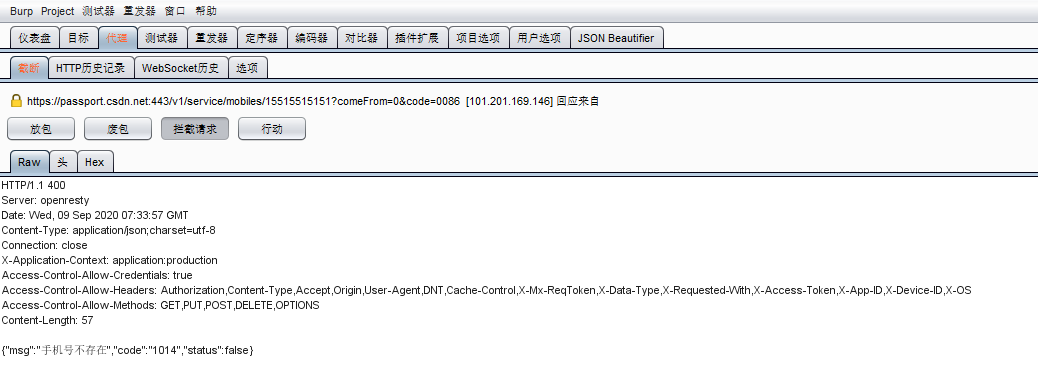
获得响应



传入手机号，进行查询



获得响应，手机号不存在。



## 实验二

### Task1

安装php及其相关依赖。

sudo apt install php7.0-mysql php7.0-curl php7.0-json php7.0-cgi php7.0 libapache2-mod-php7.0

编写hello.php，给出的代码有问题，最后一句中的name要加引号。

代码如下：

<?php

//Get current URL

function GetCurUrl()

{

$url = 'http://';

if (isset($\_SERVER['HTTPS']) && $\_SERVER['HTTPS'] == 'on') {

$url = 'https://';

}

//judge port

if ($\_SERVER['SERVER\_PORT'] != '80') {

$url .= $\_SERVER['SERVER\_NAME'] . ':' . $\_SERVER['SERVER\_PORT'] . ':' . $\_SERVER['REQUEST\_URI'];

} else {

$url .= $\_SERVER['SERVER\_NAME'] . ':' . $\_SERVER['REQUEST\_URI'];

}

return $url;

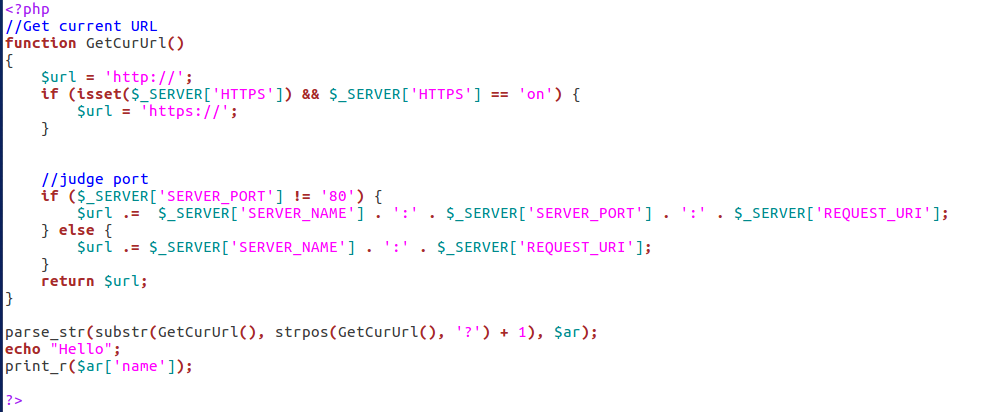
}

parse\_str(substr(GetCurUrl(), strpos(GetCurUrl(), '?') + 1), $ar);

echo "Hello";

print\_r($ar['name']);

?>



访问<http://vulnerable/hello.php?name=xxx> ，成功访问。



### Task2

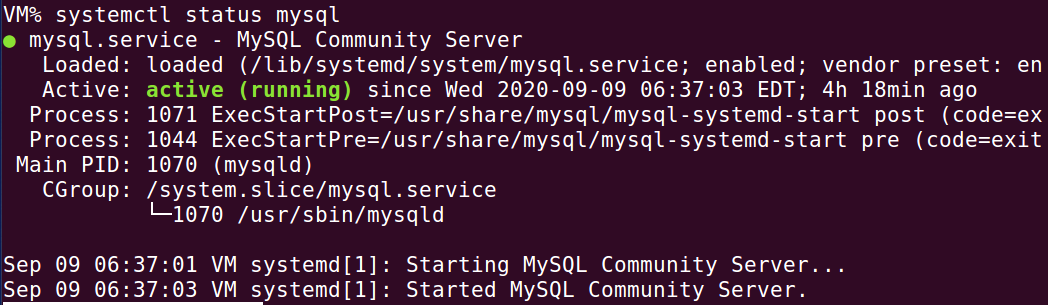
安装mysql

sudo apt-get install mysql-server mysql-client

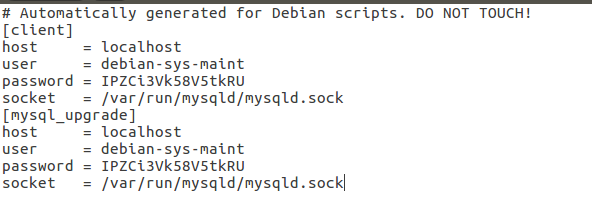
查看mysql状态是否启动

systemctl status mysql

运行良好



sudo gedit /etc/mysql/debian.cnf打开该文件，查看mysql为我们创建的的一个用户，找到用户名和密码



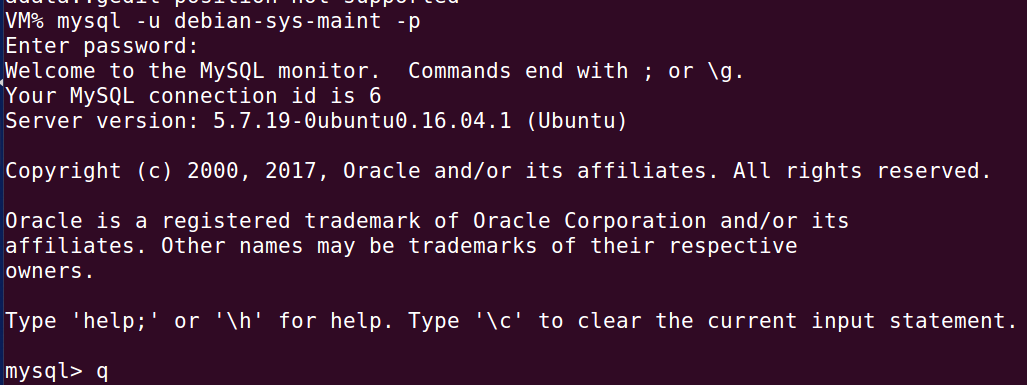
user = debian-sys-maint

password = IPZCi3Vk58V5tkRU

然后在终端输入 mysql -u debian-sys-maint -p 然后回车输入文件里显示的密码（之后编写php文件对数据库进行访问的时候也需要这个用户名和密码才能创建与数据库的连接）

-u选项跟的是username，这里的debian-sys-maint用户就是前面创建的默认用户。

-p即输入密码。



进入mysql操作界面后，创建接下来需要使用的数据库以及相关的表。

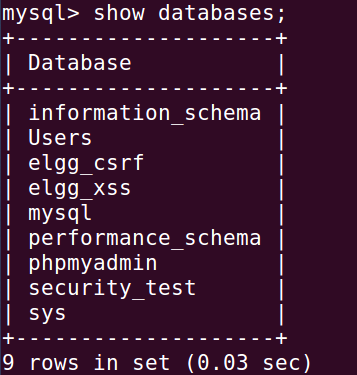
mysql命令以分号结尾 必须要带分号。

1）mysql>create database security\_test;



2）mysql>show databases; 查看是否创建成功

可以看到我们刚刚创建的database



3）mysql>use security\_test;进入创建好的数据库



4）创建用户信息表

输入

create table user\_info(

userid int not null primary key auto\_increment,

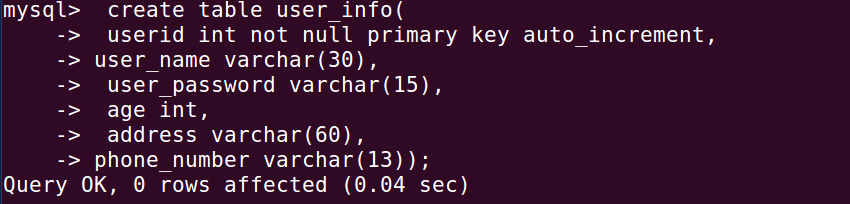
user\_name varchar(30),

user\_password varchar(15),

age int,

address varchar(60),

phone\_number varchar(13));



5）创建用户好友列表

create table user\_friends(

friendid int not null primary key auto\_increment,

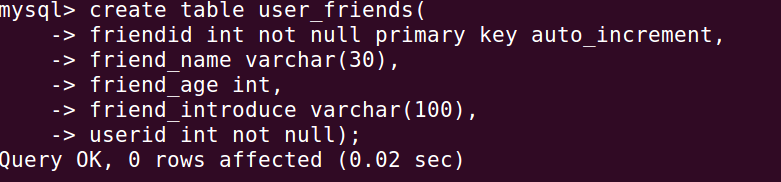
friend\_name varchar(30),

friend\_age int,

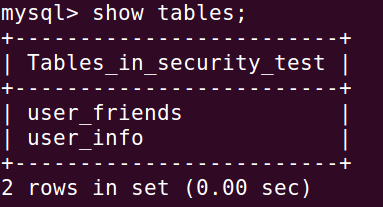
friend\_introduce varchar(100),

userid int not null);

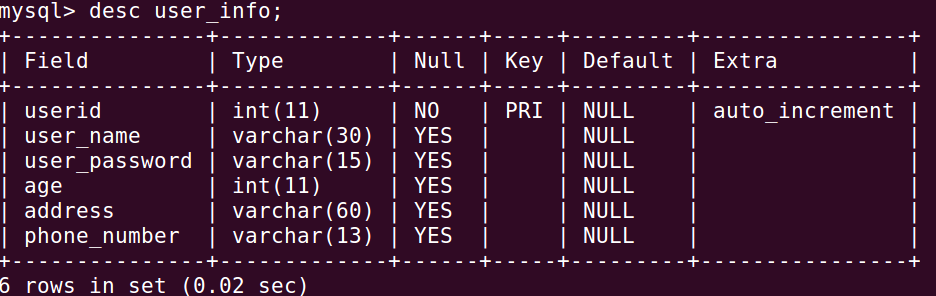
实验手册给错了，这里应该是一整个括号，包含起所有参数。



6）mysql>show tables;查看表信息，



mysql>desc user\_info;查看表字段信息



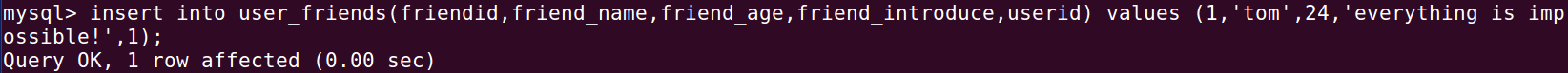
7）为两个表插入数据，插入格式如下，可自行多插入几条数据

mysql> insert into user\_info values (1,'bob','123456',24,'china','14786432587');

实验手册给错了，没有'nothing is impossible!'这一项。

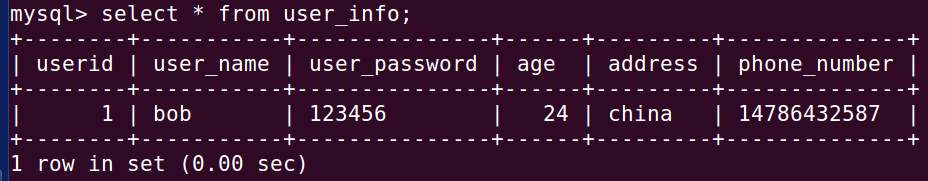


mysql> insert into user\_friends(friendid,friend\_name,friend\_age,friend\_introduce,userid) values (1,'tom',24,'everything is impossible!',1);

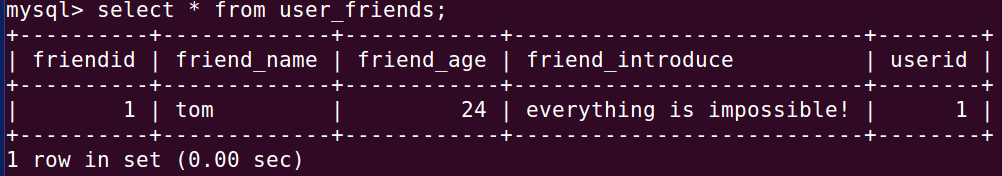


8）查看表格信息，以table名作为参数。

select \* from user\_info;



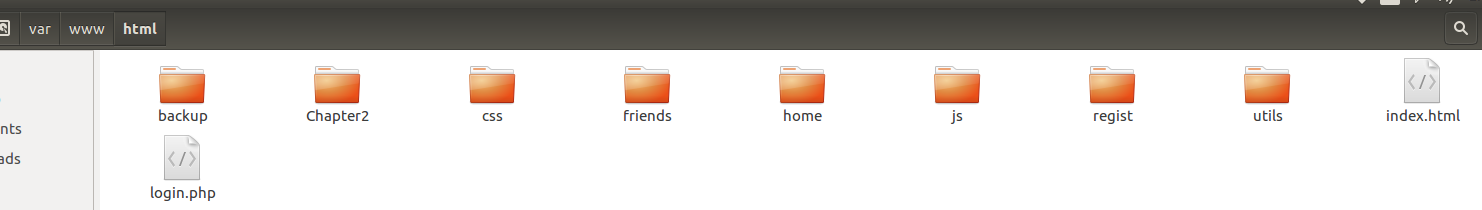
select \* from user\_friends;



### Task3 总体测试

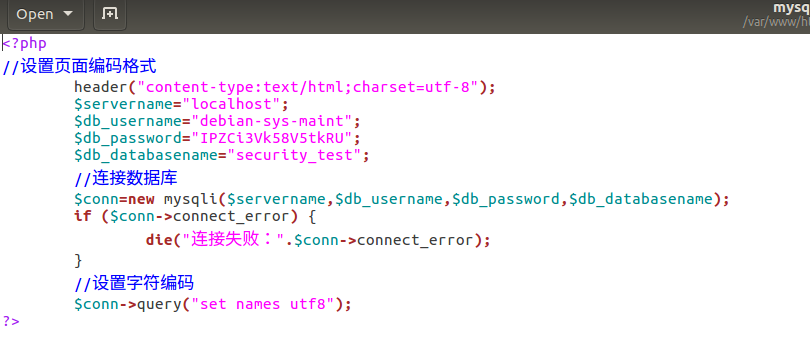
步骤 1：将项目解压到apache的启动目录（/var/www/html）

必须是根目录，否则css会因为路径问题无法加载。



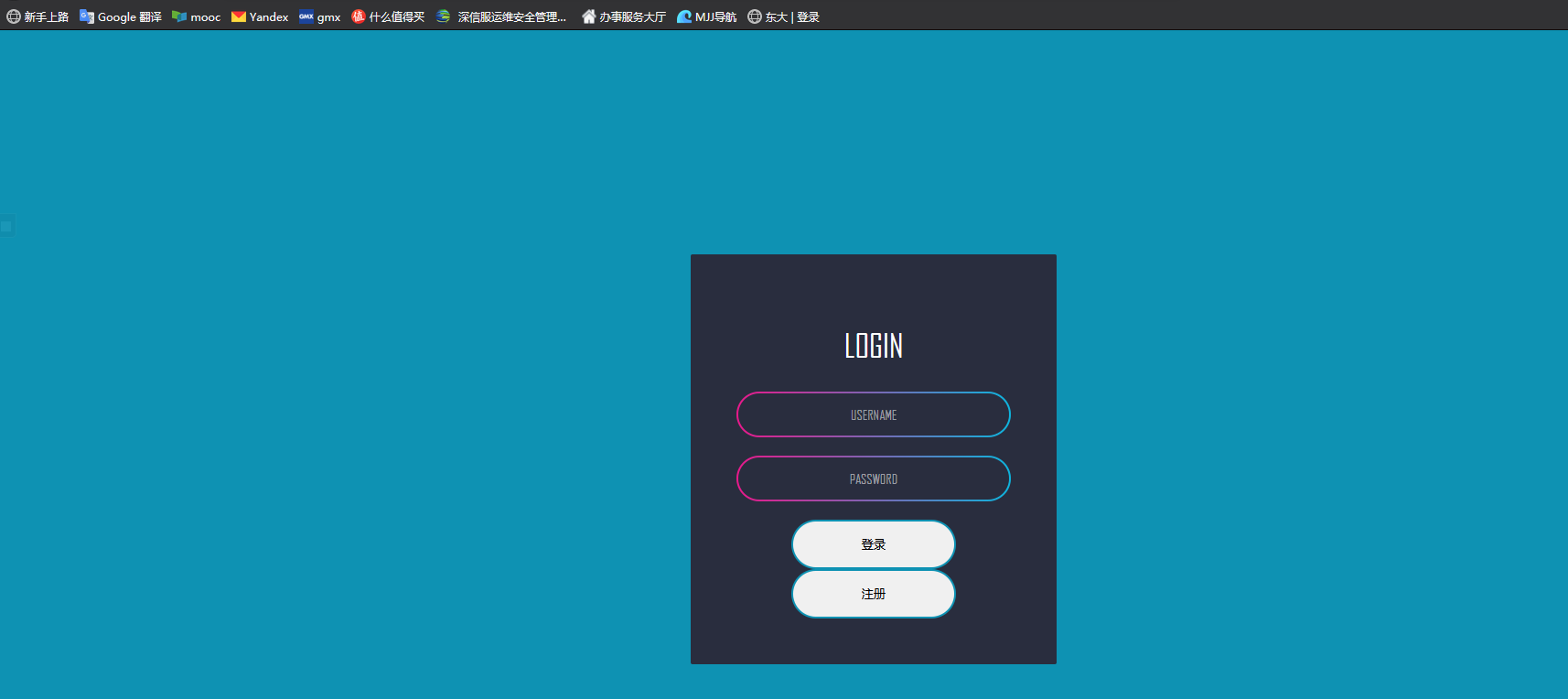
步骤 2：打开项目的utils文件夹下的mysqlBase.php，核对虚拟机上的mysql用户名与密码还有数据库名称，匹配不上会导致连接失败

核对是一样的

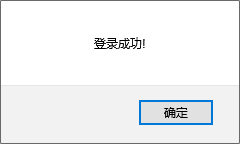


步骤 2：使用主机浏览器访问地址：http://vulnerable ，即可测试登录注册，修改个人信息，添加删除好友列表等等

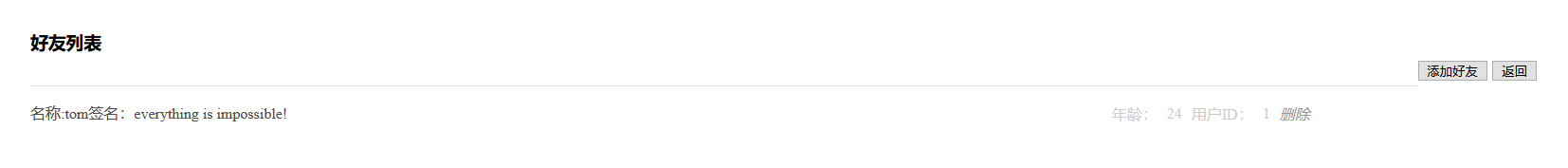
登录界面



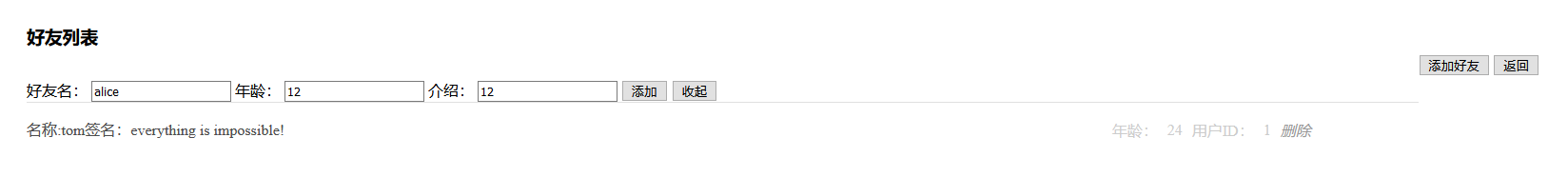
登录信息选择自己写进表（user\_info）的账号密码，即上述插入的bob，123456

查看好友列表



添加好友





更新后的好友列表

