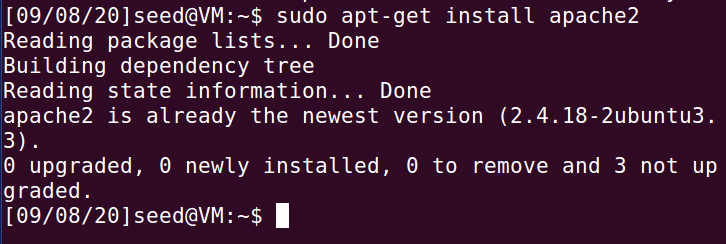
57118133 钟杰

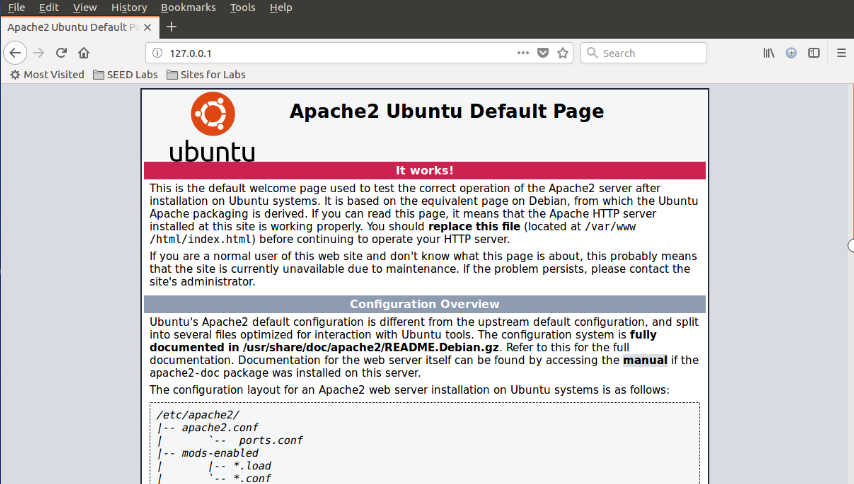
实验一 HTTP基础

任务一：安装apache服务器 并用简单页面验证

步骤 1：在虚拟机中打开terminal终端窗口，输入sudo apt-get install apache2

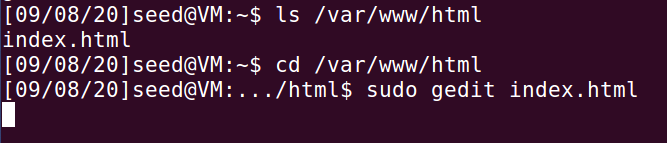


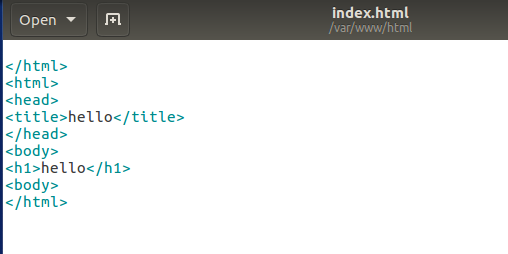
步骤 2：Apache安装完成后，默认的网站根目录是”var/www/html”，在网站根目录路径下有一个index.html文件，在本机或虚拟机浏览器中输入”127.0.0.1”就可以打开该页面。



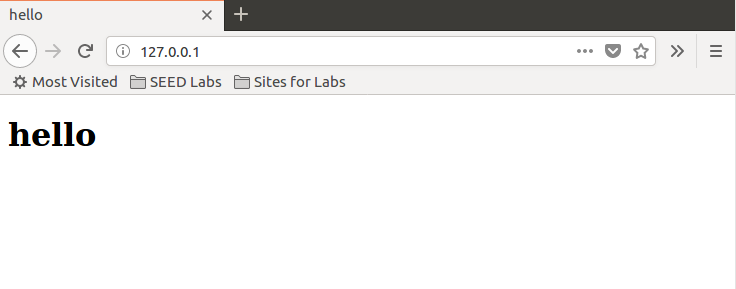
步骤 3：

1.cd /var/www/html

2.使用sudo gedit index.html指令打开index.html并进行编写

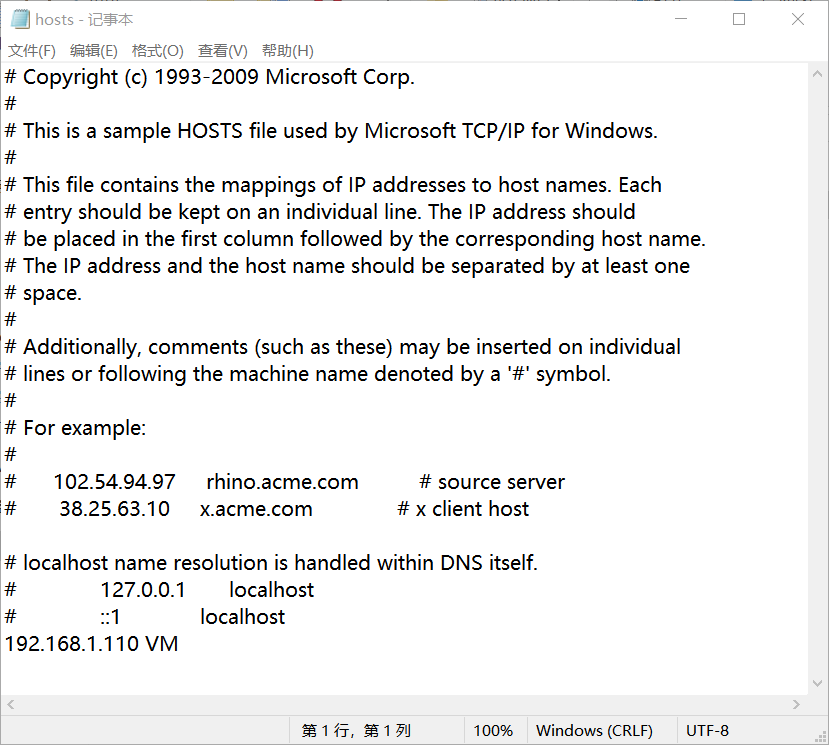


步骤 4：修改后使用浏览器登录127.0.0.1，页面更改为新主页。



任务二：通过host文件解析名称

步骤 1： 在windows主机中找到hosts文件记事本打开，修改hosts文件加入虚拟机ip地址与主机名vulnerable并保存（C:\Windows\System32\drivers\etc）



任务三：编写HTTP客户端，使用http库检索站点的主页

步骤1：windows主机中输入curl+虚拟机ip地址可查看编写的index文件内容



任务四：编写HTTP客户端以使用套接字检索站点的主页，代码如下：

步骤1：在主机创建c语言程序，写入如下代码

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

#include <winsock2.h>

#include<time.h>

#pragma comment(lib,"ws2\_32.lib")

#pragma warning（disable：4996）

void ReadPage(char\* host)

{

WSADATA data;

//winsock版本2.2

int err = WSAStartup(MAKEWORD(2, 2), &data);

if (err)

return;

//用域名获取对方主机名

struct hostent \*h = gethostbyname(host);

if (h == NULL)

return;

//IPV4

if (h->h\_addrtype != AF\_INET)

return;

struct in\_addr ina;

//解析IP

memmove(&ina, h->h\_addr, 4);

LPSTR ipstr = inet\_ntoa(ina);

//Socket封装

struct sockaddr\_in si;

si.sin\_family = AF\_INET;

si.sin\_port = htons(80);

si.sin\_addr.S\_un.S\_addr = inet\_addr(ipstr);

int sock = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(sock, (SOCKADDR\*)&si, sizeof(si));

if (sock == -1 || sock == -2)

return;

//发送请求

char request[1024] = "GET /?st=1 HTTP/1.1\r\nHost:";

strcat(request, host);

strcat(request, "\r\nConnection:Close\r\n\r\n");

int ret = send(sock, request, strlen(request), 0);

//获取网页内容

FILE \*f = fopen("recieved.txt", "w");

int isstart = 0;

while (ret > 0)

{

const int bufsize = 1024;

char\* buf = (char\*)calloc(bufsize, 1);

ret = recv(sock, buf, bufsize - 1, 0);

printf(buf);

fprintf(f, "%s", buf);

free(buf);

}

fclose(f);

closesocket(sock);

WSACleanup();

printf("读取网页内容成功，已保存在recieved.txt中\n");

return;

}

void main() {

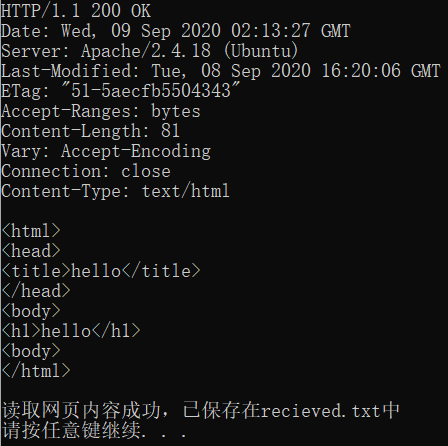
char\* str = "VM";

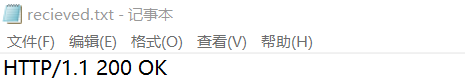
ReadPage(str);

system("pause");

}

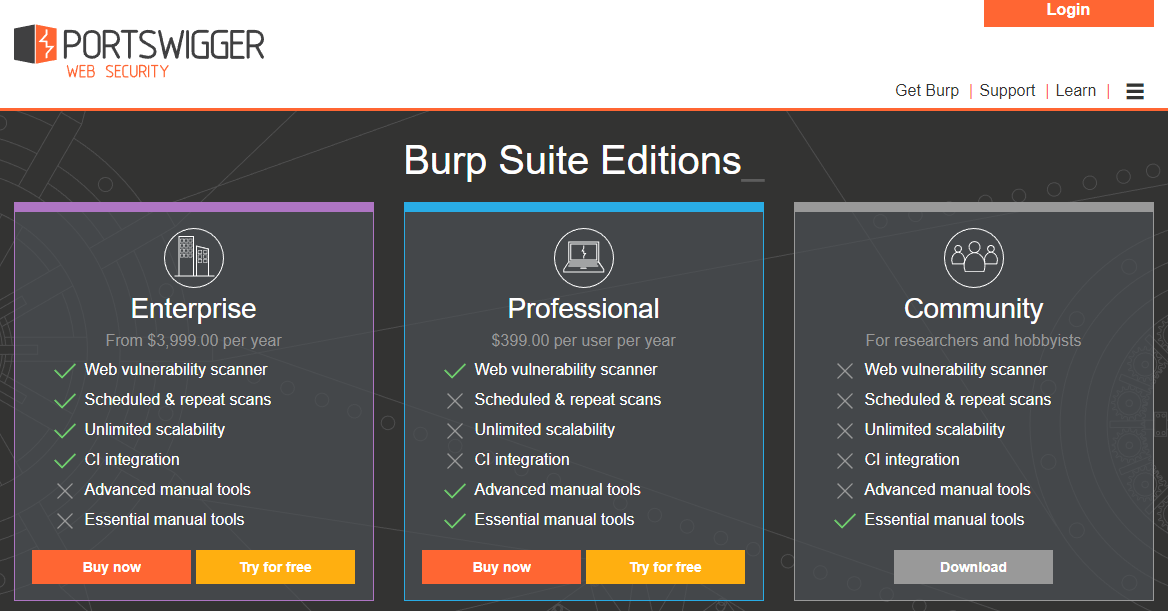
步骤2：执行该文件，网页定向正确





任务五：下载软件Burp Suite并访问网站查看请求与响应的信息

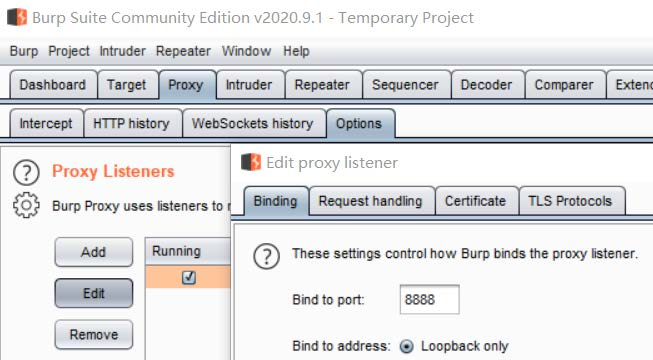
步骤 1：从<https://portswigger.net/burp>网站中下载Comuunity版本（需配置jdk环境）



步骤 2：对测试浏览器Chrome进行代理设置,地址设为127.0.0.1,端口修改为8888



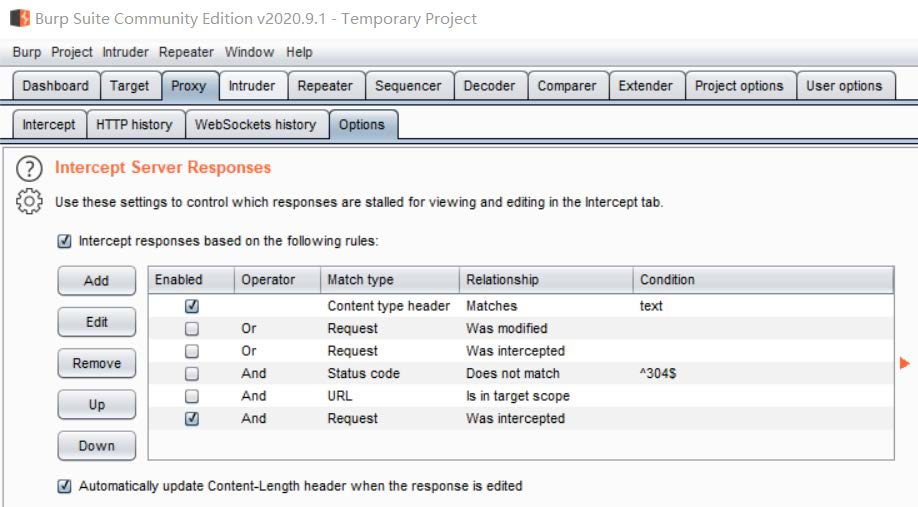
步骤 3：打开Burp Suite界面，设置Proxy代理，端口改为8888（需安装CA安全证书）



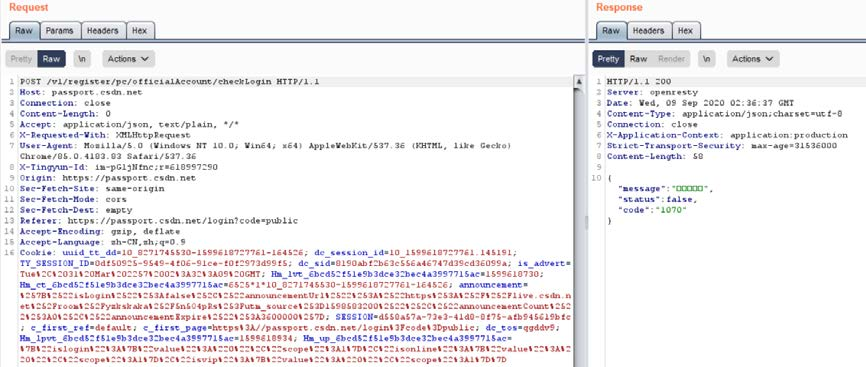
步骤 4：使用浏览器打开my.seu.edu.cn查看拦截情况（使用open browser按钮打开浏览器，输入网址后下图会有类似的信息，如果不是对应的地址，请点击forward/drop选择找到响应的my.seu.edu.cn的信息）



步骤 5：



步骤 6：测试CSDN通过发送验证码找回密码功能，查看Request和Response功能(网站进行访问时需要点击forward按钮才能不断发送请求与接收响应，在测试CSDN之前需要对网页进行多次访问，因此可以先关闭拦截，点击Intercept is on按钮进行关闭，在需要拦截时再打开)



实验二 使用PHP和Mysql搭建一个简单的站点

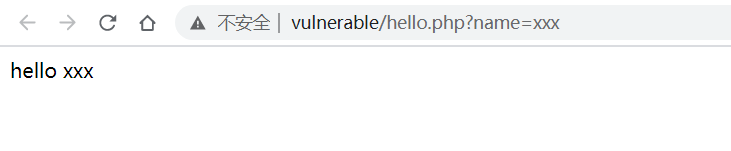
任务一：在虚拟机中安装PHP（使用以前的Apache安装），编写一个脚本以回显URL中的参数。

步骤 2:编写hello.php，使用命令sudo nautilus以管理员方式打开文件管理器，将该文件放入var/www/html，删除原来编写的index.html文件





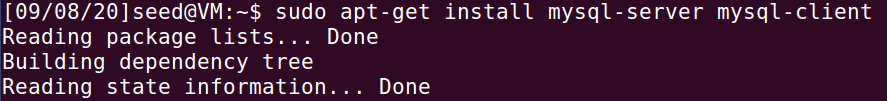
步骤3：在主机中打开浏览器，输入链接<http://vulnerable/hello.php?name=xxx>

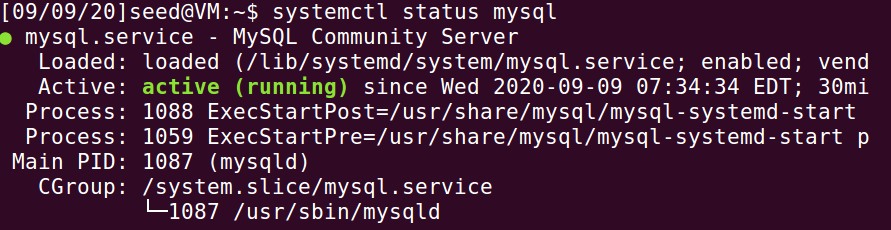


任务二：安装mysql服务

步骤1：install Mysql

在终端输入sudo apt-get install mysql-server mysql-client

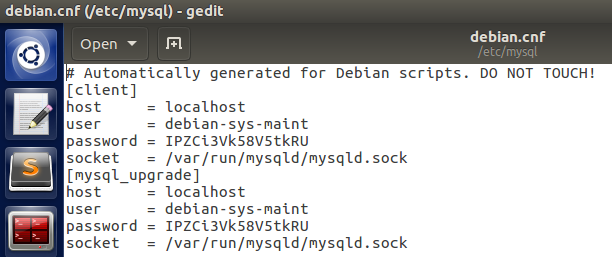




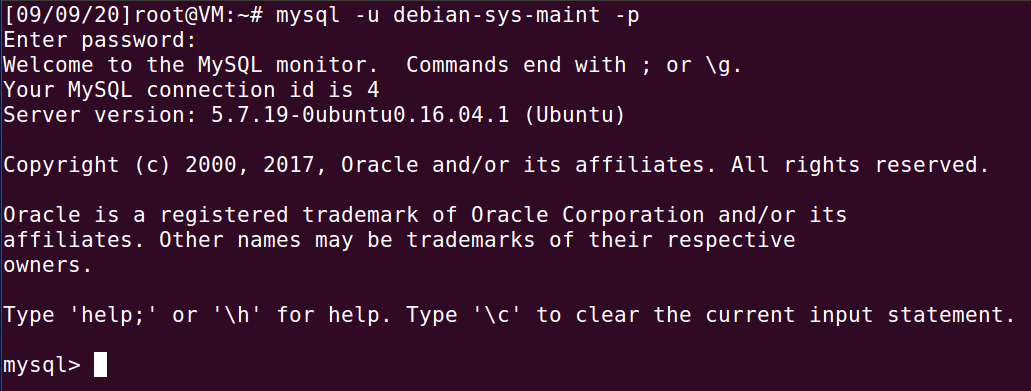
步骤 2：systemctl status mysql查看mysql状态是否启动，未激活则

systemctl start mysql进行启动

步骤 3：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



步骤4：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，大家也可以自行百度创建自己的mysql用户）



步骤5：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

1. 创建数据库和表(mysql命令以封号;结尾,必须要带封号)
2. mysql>create database security\_test;



1. mysql>use security\_test;进入创建好的数据库



4）创建用户信息表

mysql> create table user\_info(

-> userid int not null primary key auto\_increment,

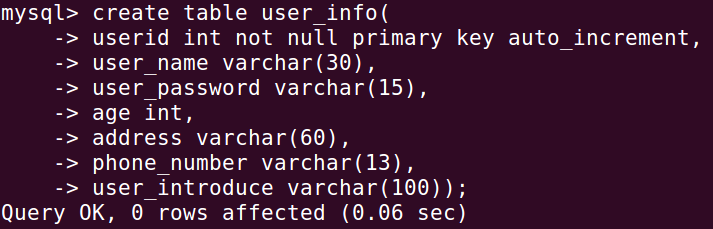
-> user\_name varchar(30),

-> user\_password varchar(15),

-> age int,

-> address varchar(60),

-> phone\_number varchar(13));



5）创建用户好友列表

mysql> create table user\_friends(

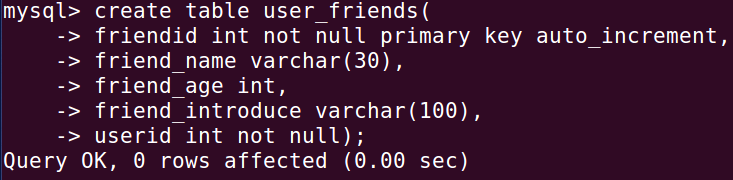
-> friendid int not null primary key auto\_increment,

-> friend\_name varchar(30),

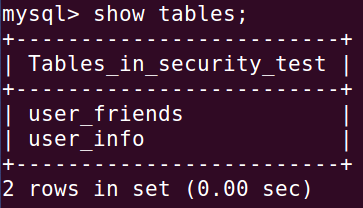
-> friend\_age int,

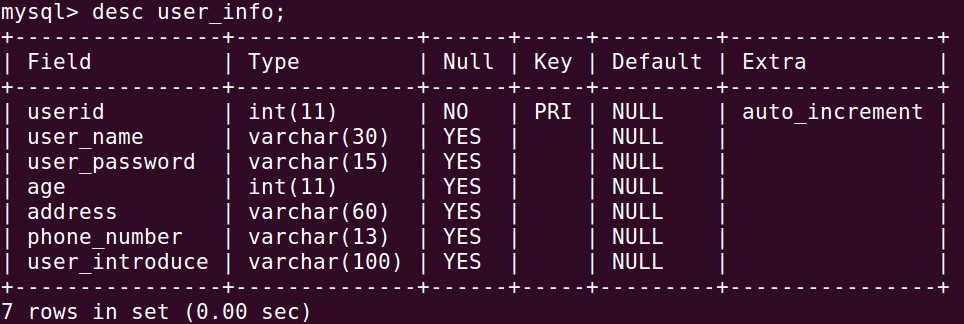
-> friend\_introduce varchar(100)),

-> userid int not null;



1. 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','nothing is impossible!');

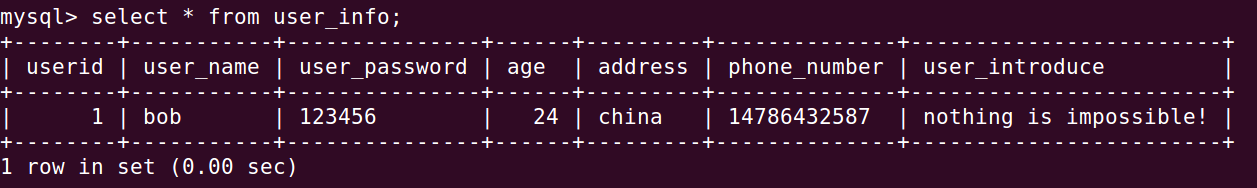
mysql> insert into user\_friends(friendid,friend\_name,friend\_age,friend\_introduce,userid) values (1,'tom',24,'everything is impossible!',1);

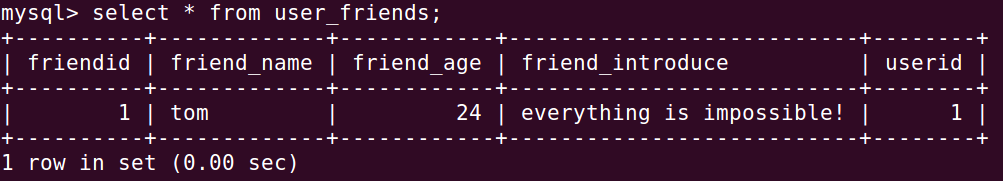




8）查看表格信息

mysql> select \* from 表名

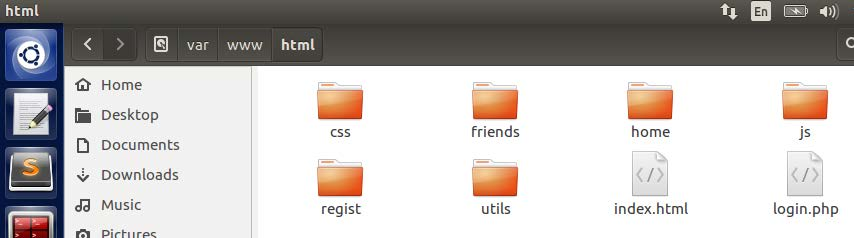




任务三：测试运行一个简易的项目模板

总体测试步骤：

步骤 1：将项目解压到apache的启动目录（/var/www/html）,移除该文件夹原来编写的测试文件（自己编写的同学们略过这步处理）



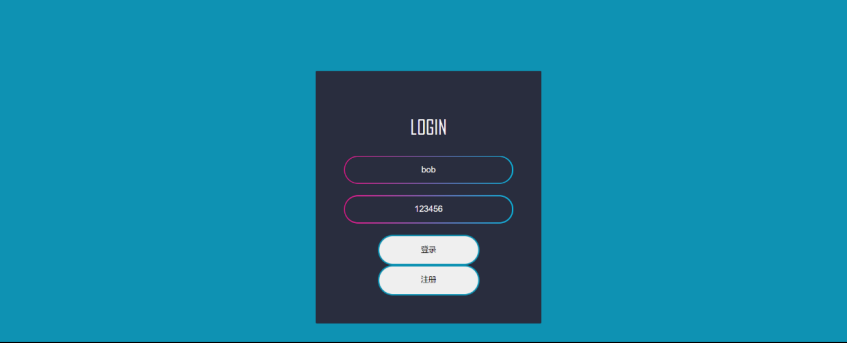
步骤 2：打开项目的utils文件夹下的mysqlBase.php，核对虚拟机上的mysql用户名与密码还有数据库名称，匹配不上会导致连接失败（自己编写的同学们略过这步处理）



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

注册界面



个人信息界面



点击返回，进入好友列表界面

IMG_256

点击添加好友