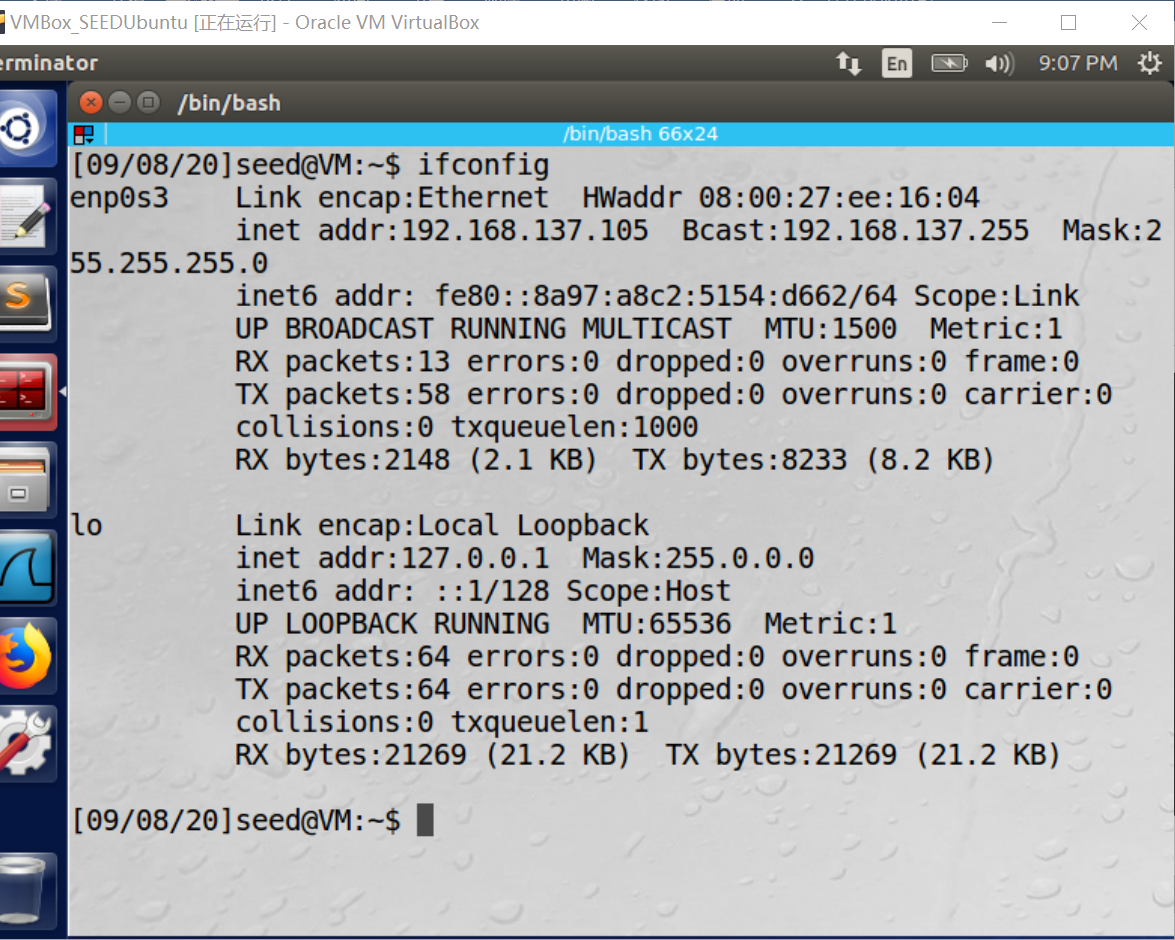
**HTTP基础实验报告**

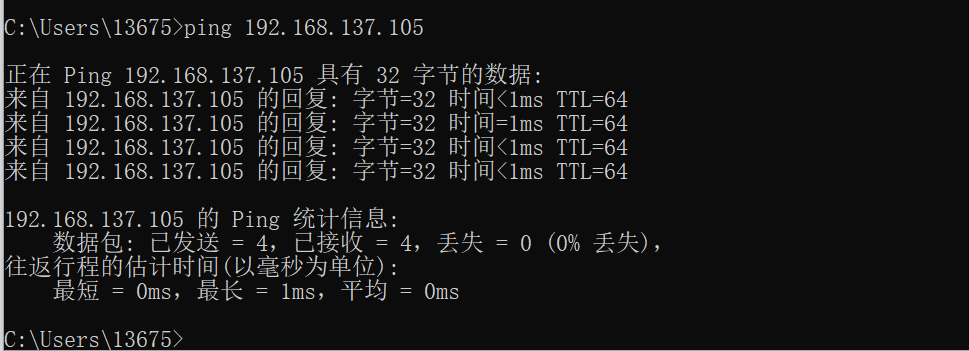
57118112-王怡乐

**Task0:搭建简单的Web站点**

将虚拟机的网络更改为桥接模式，并在虚拟机上查询虚拟机的IP地址：

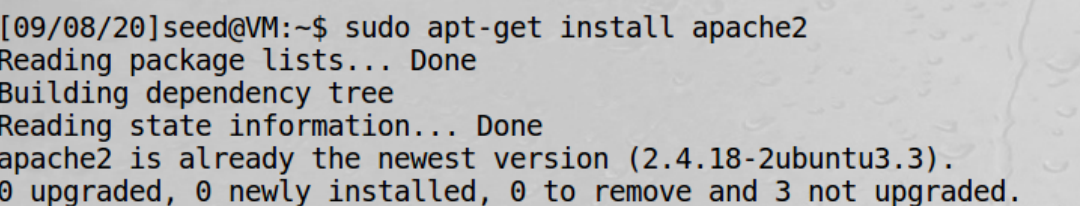


在主机上ping 虚拟机，成功：

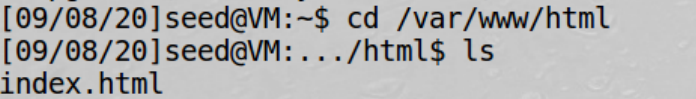


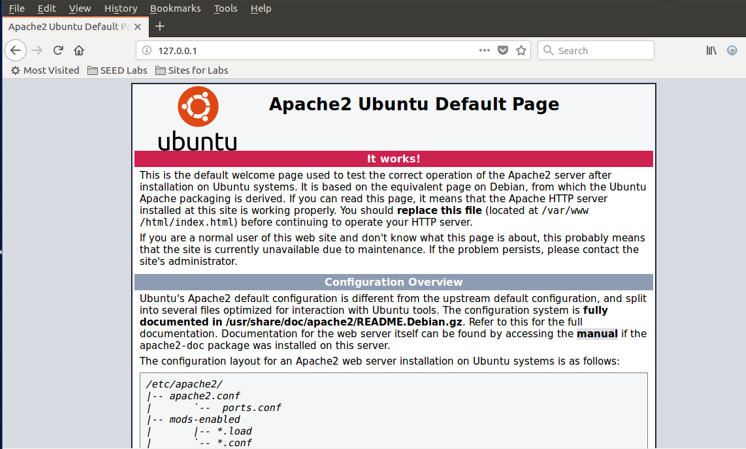
**Task1:安装apache服务器，并用简单页面验证**

1、在虚拟机中打开terminal终端，输入sudo apt-getinstall apache，安装apache服务器：



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：





3、使用sudo gedit index.html指令打开index.html文件并进行编写：



内容如下：

<html>

<head>

<title>hello</title>

</head>

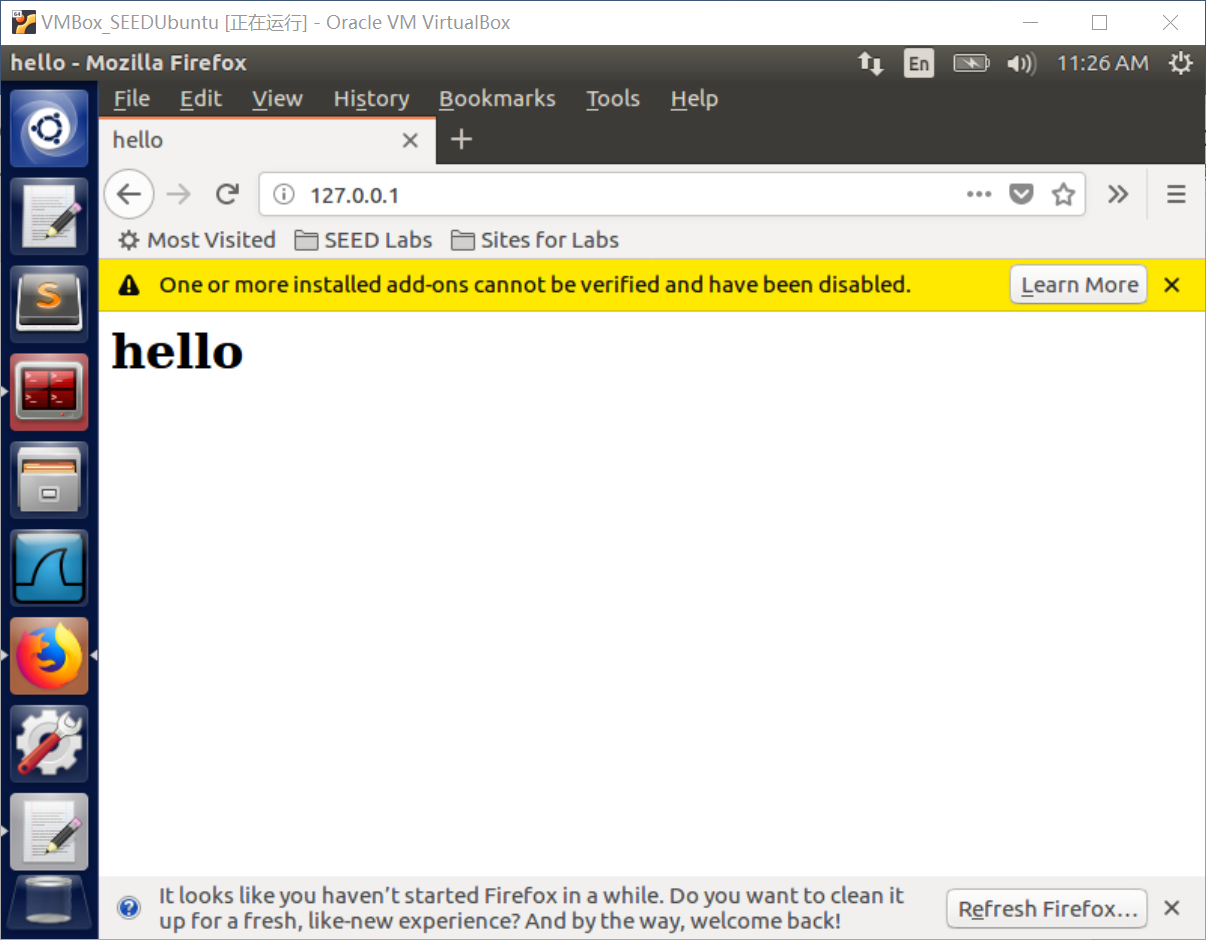
<body>

<h1>hello</h1>

<body>

</html>

4、修改后使用浏览器登录127.0.0.1，页面更新为新主页：

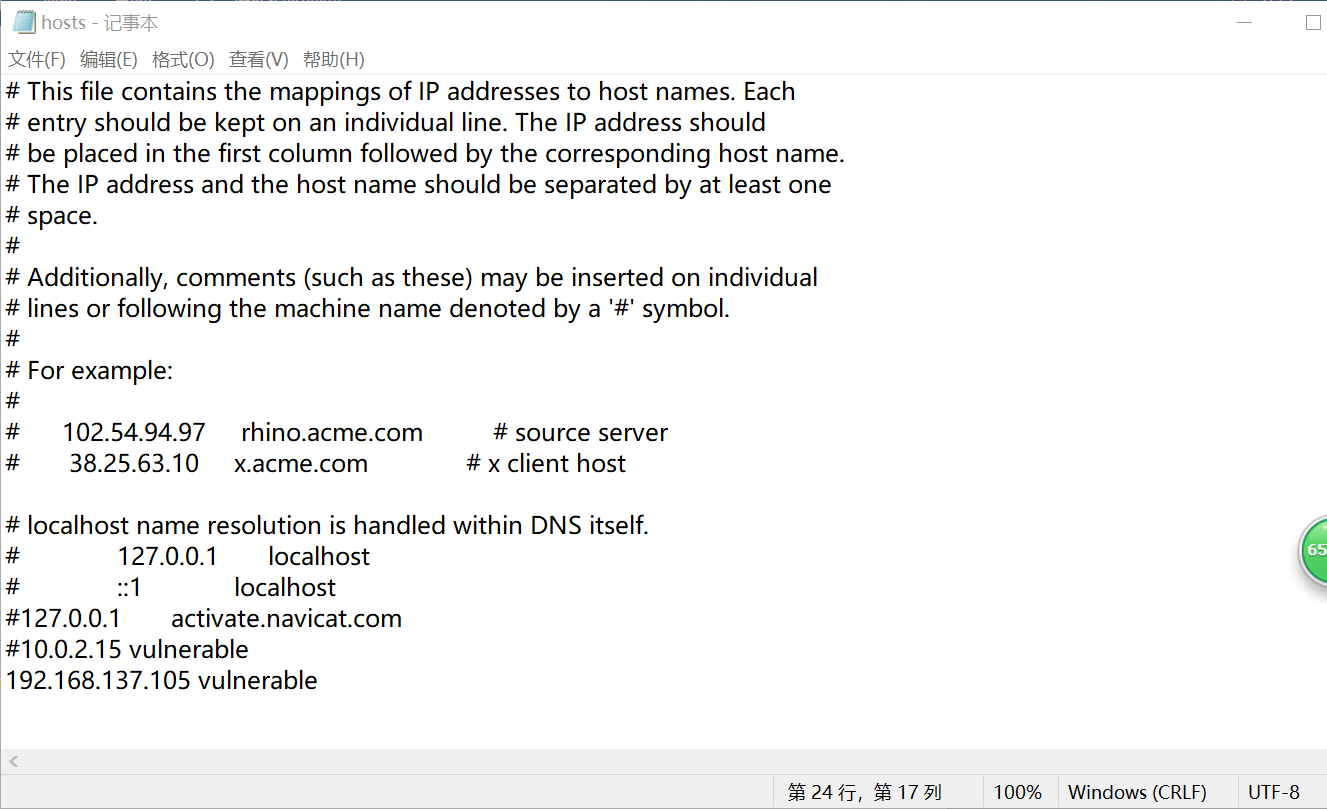


**Task2:通过host文件解析名称**

1、以管理员身份打开记事本，找到C:\Windows\System32\drivers\etc\hosts并打开：

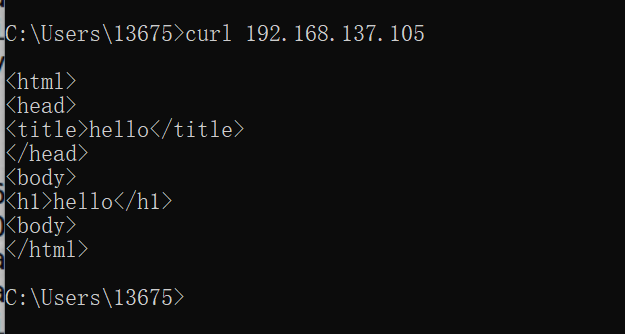


2、在文件最后加入虚拟机的IP地址并保存文件：

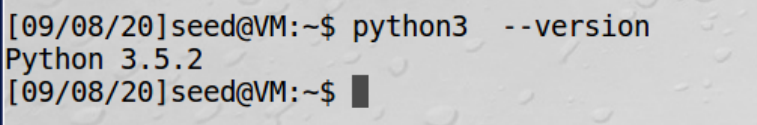


**Task3:编写HTTP客户端，使用HTTP库检索站点的主页**

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



2、虚拟机中输入python3 –version，确认虚拟机的python版本为3.5：



**Task4:编写HTTP客户端以使用套接字检索站点的主页**

1. 在主机创建c语言程序，代码如下：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

#include <winsock2.h>

#include<time.h>

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

void ReadPage(const 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;

}

int main() {

const char\* str = "vulnerable";

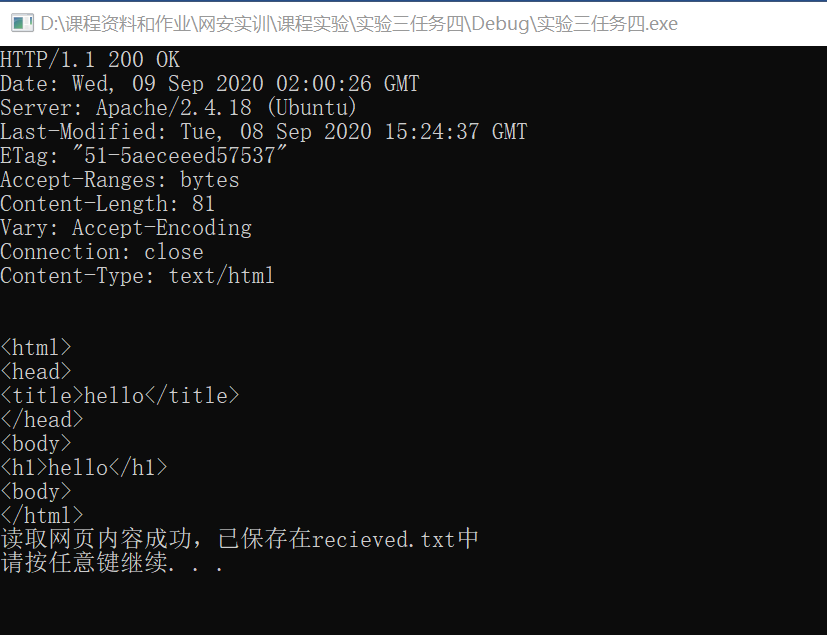
ReadPage(str);

system("pause");

return 0;

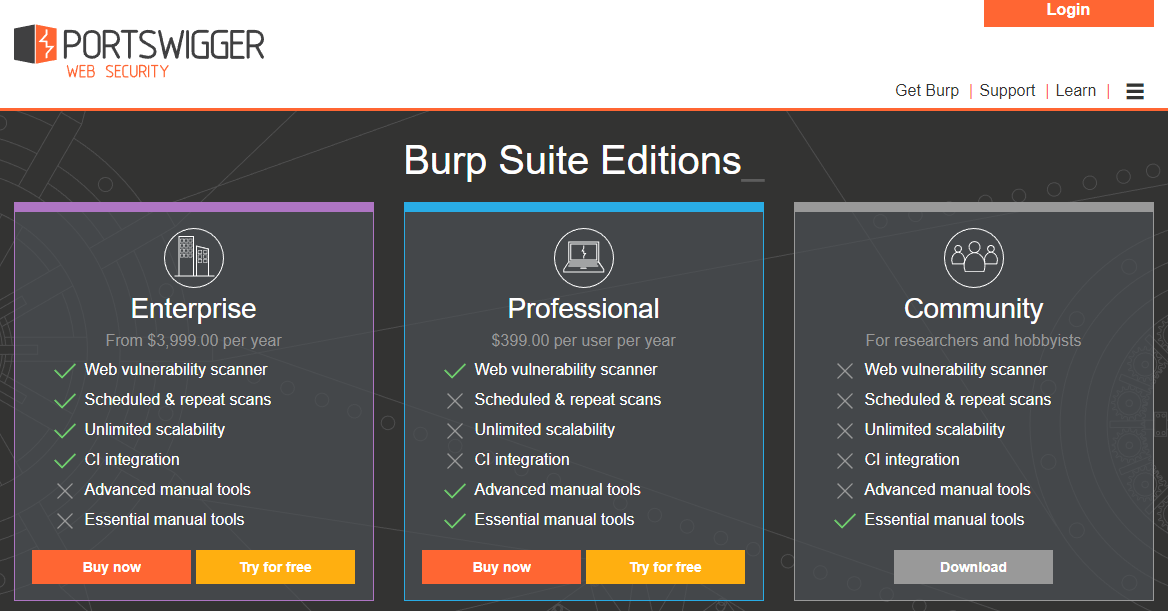
}

2、执行该文件，网页能够正确定向：



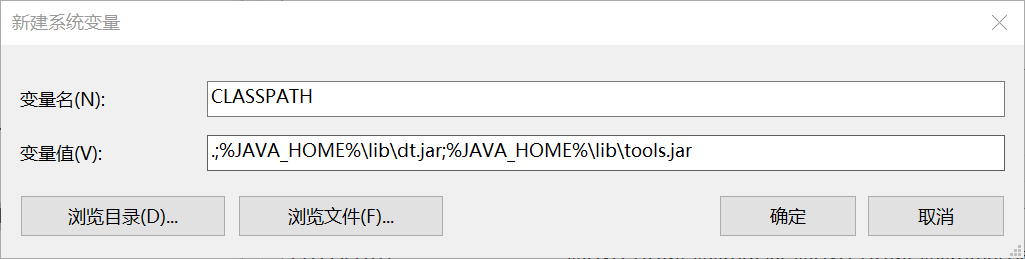
**Task5:下载软件Burp Suite并访问网站查看请求与响应的信息**

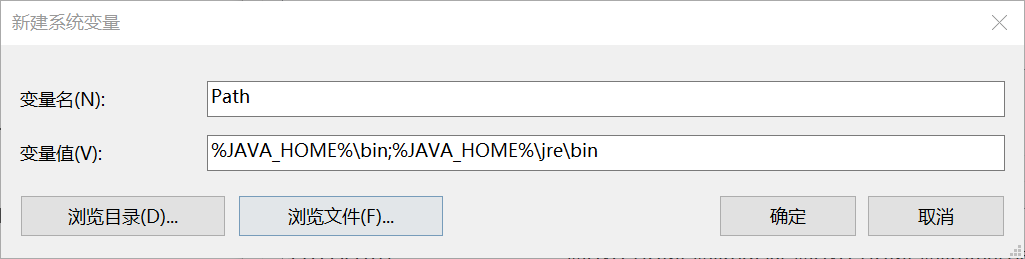
1、从<https://portswigger.net/burp>网站中下载Comuunity版本：



下载并安装jdk，配置环境变量：







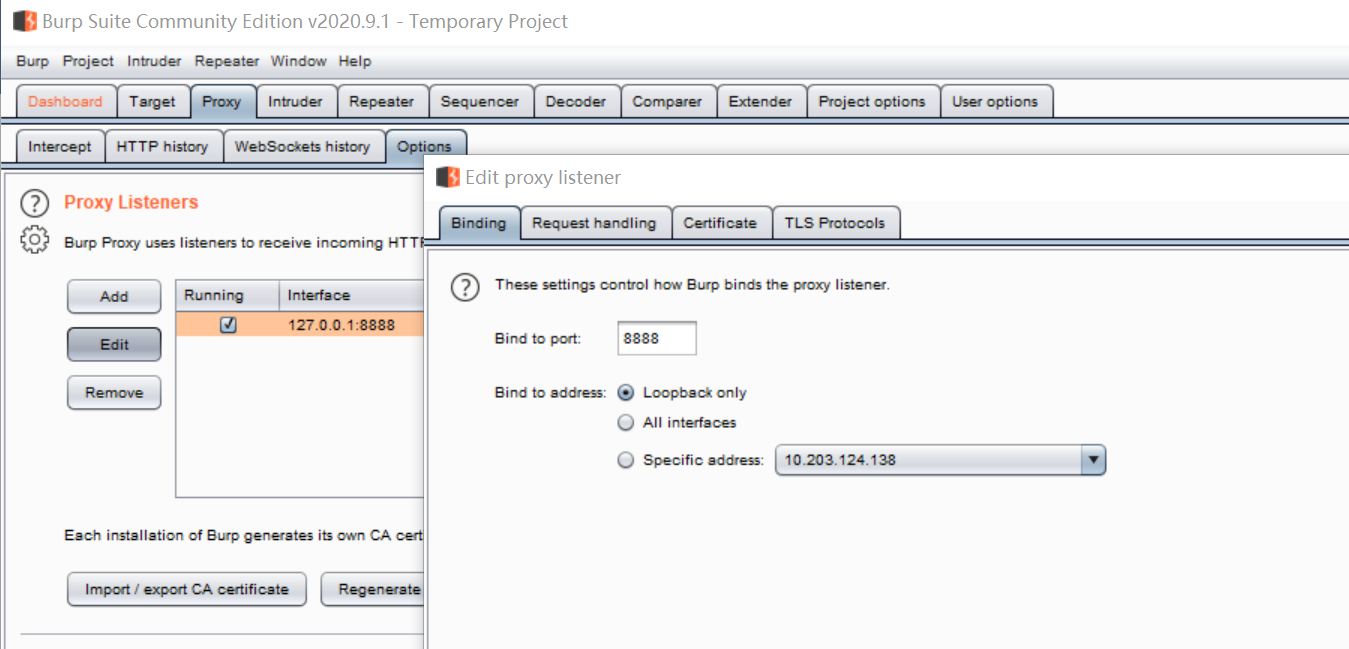
配置成功：



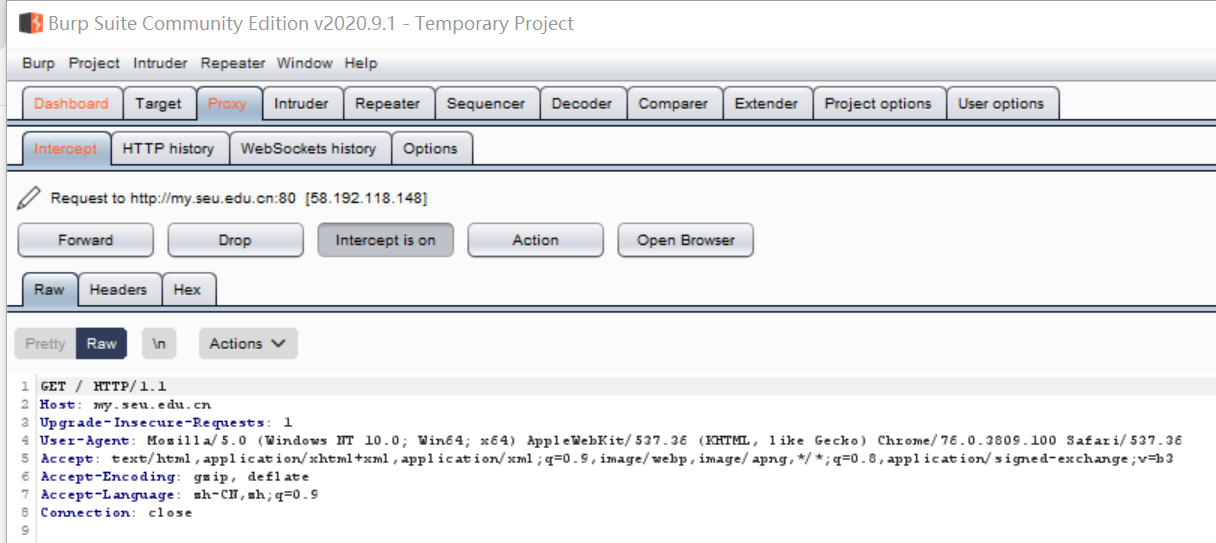
2、用open brouse打开Chrome浏览器，并对浏览器进行代理设置，地址设为127.0.0.1,端口修改为8888：



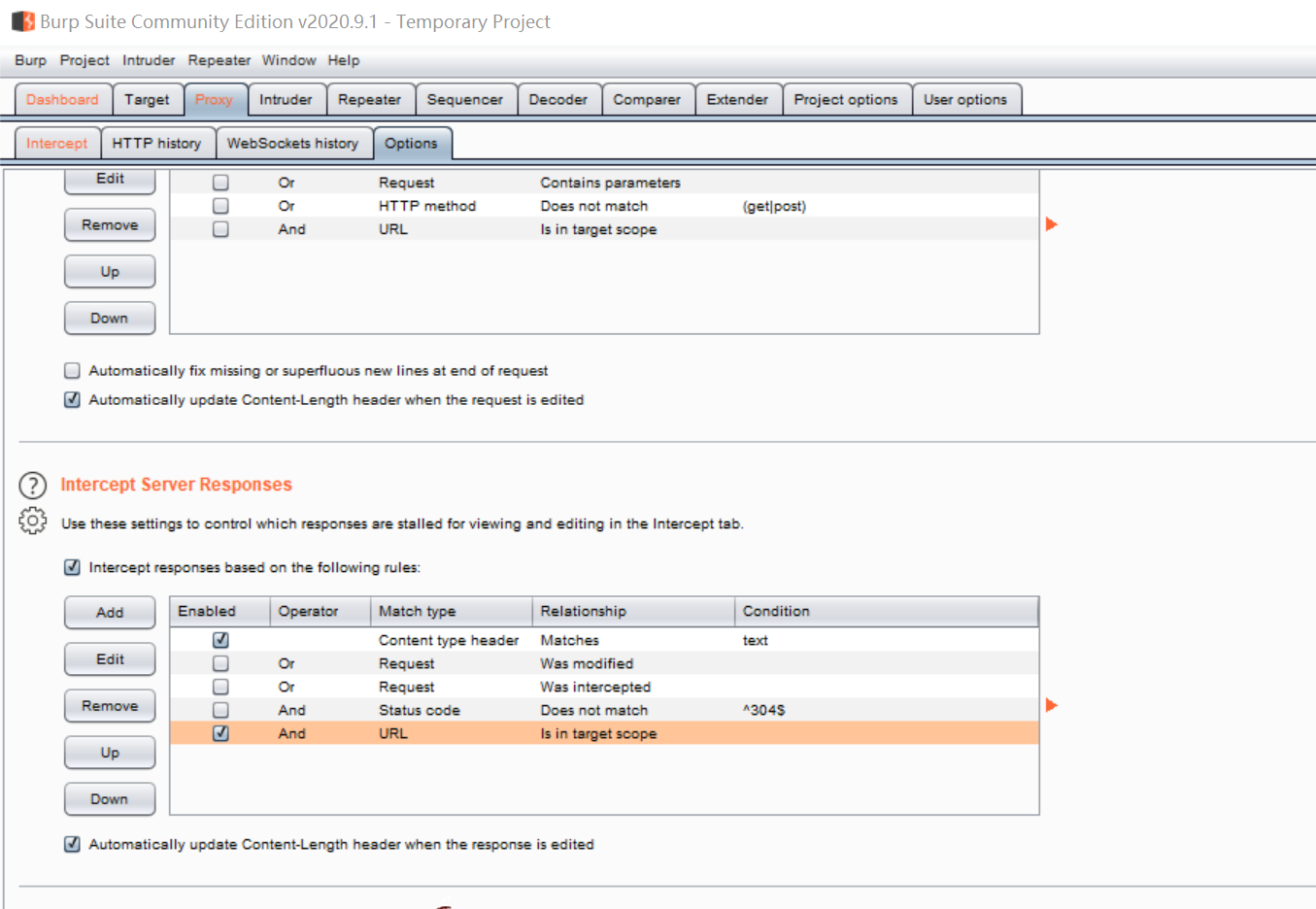
3、打开Burp Suite界面，设置Proxy代理，端口改为8888：



4、使用浏览器打开my.seu.edu.cn查看拦截情况：



5、配置拦截服务器响应的参数：

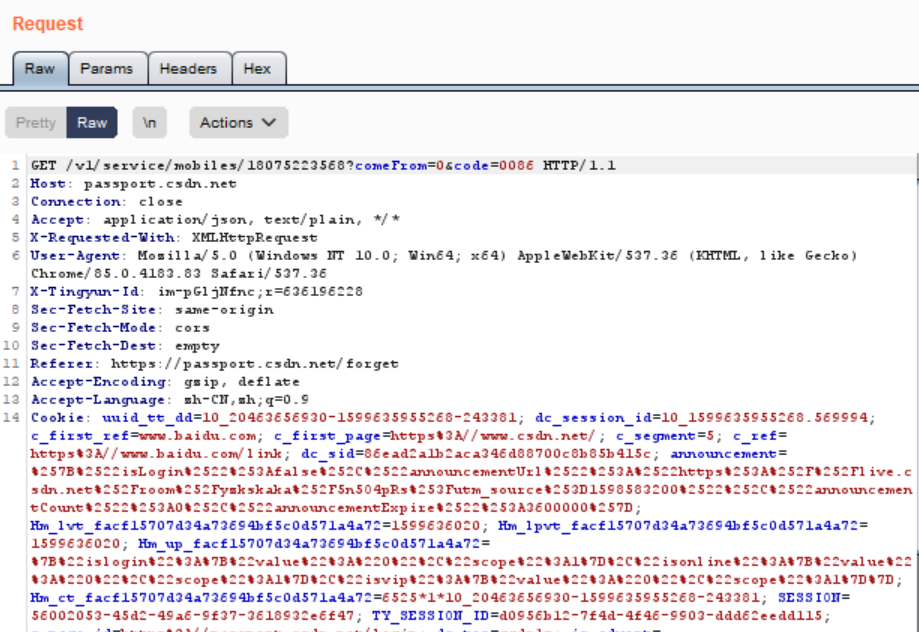


6、测试CSDN通过发送验证码找回密码功能，查看Request和Response功能。

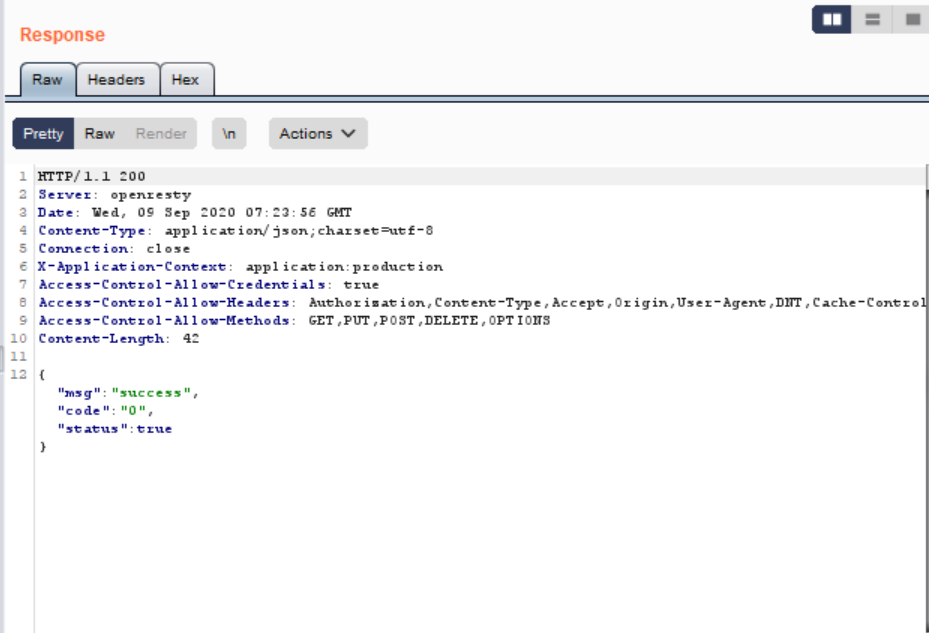
在CSDN登录页面点击忘记密码 ，出现如图:



点击获取验证码后，用burp进行拦截请求：



点击Action——>Send to Repeater ——>切换到reqpeater选项面板——>点击send发送请求——>得到响应:



**实验体会：**

本次实验主要进行了一些对于HTTP方面的操作，例如编写html文件，编写HTTP客户端等。HTTP协议在生活中应用非常广泛，但我却对它的原理知之甚少。这次的实验加深了我对HTTP的理解，这对进一步学习HTTP来说是一个较好的开端。