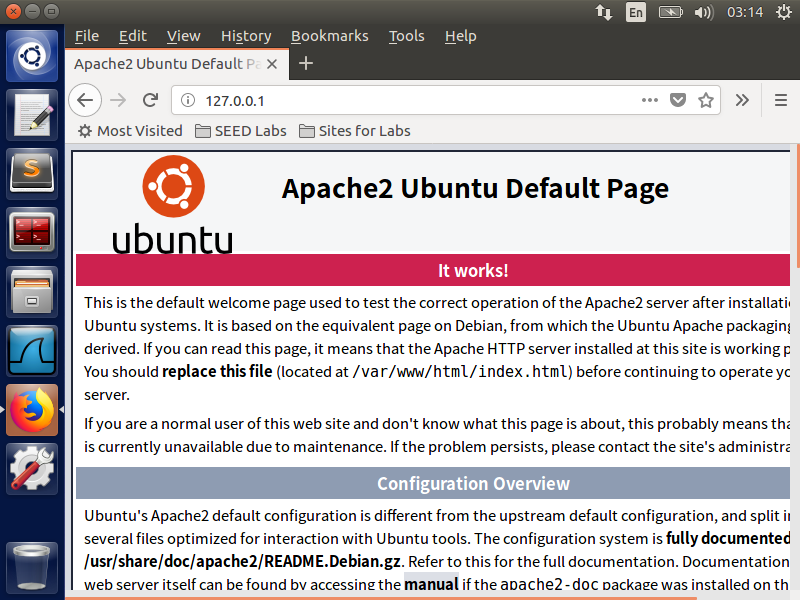
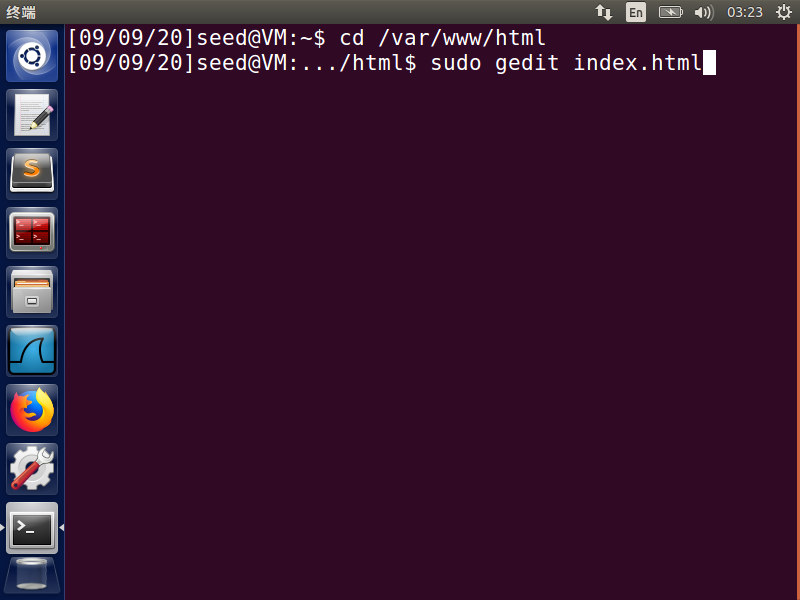
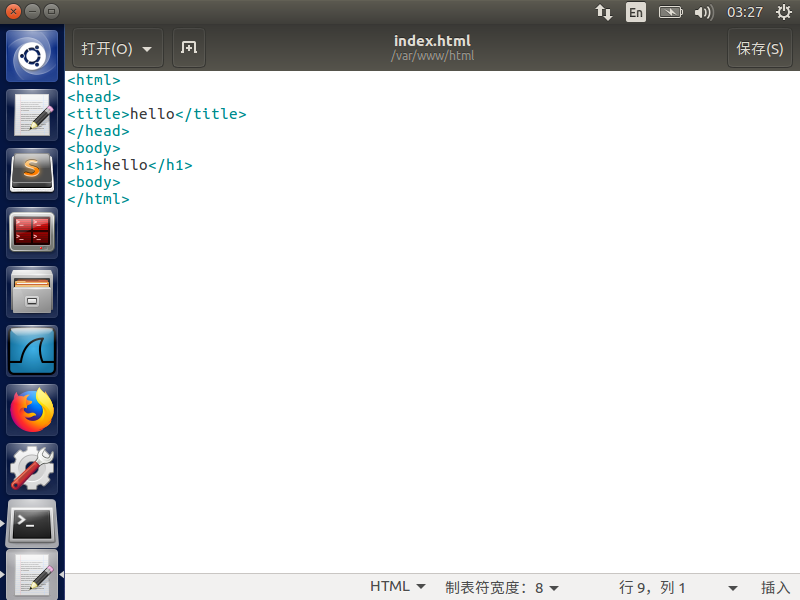
HTTP基础

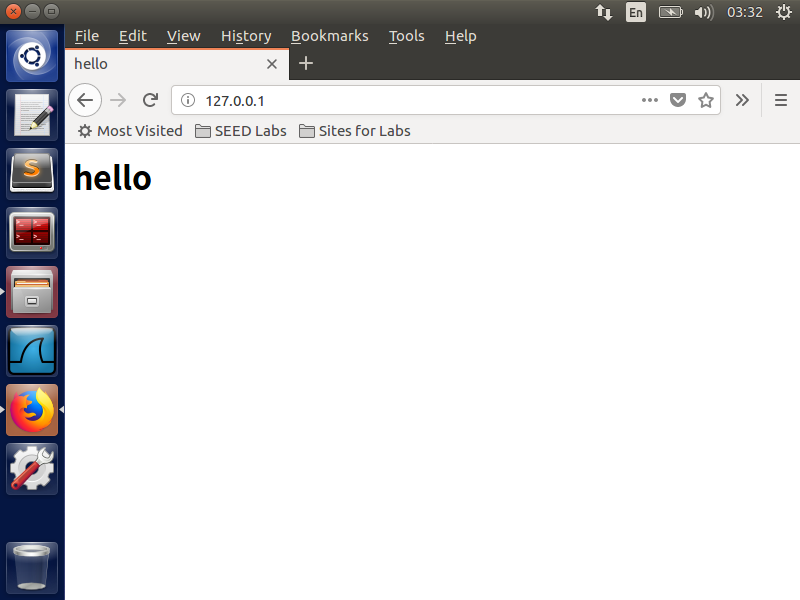
姓名：严佳豪，学号：57118136

**任务一：安装apache服务器 并用简单页面验证**

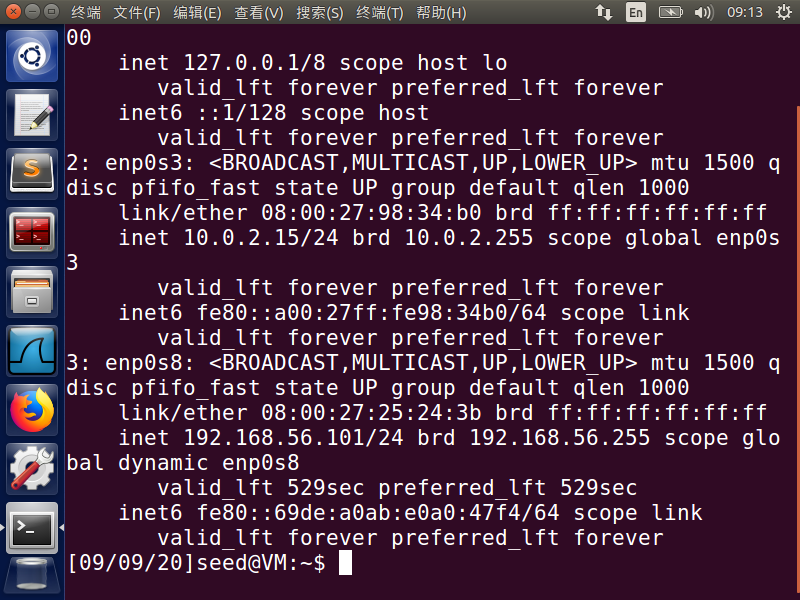
 修改前的index.html的界面如下

 在终端中输入以下命令：

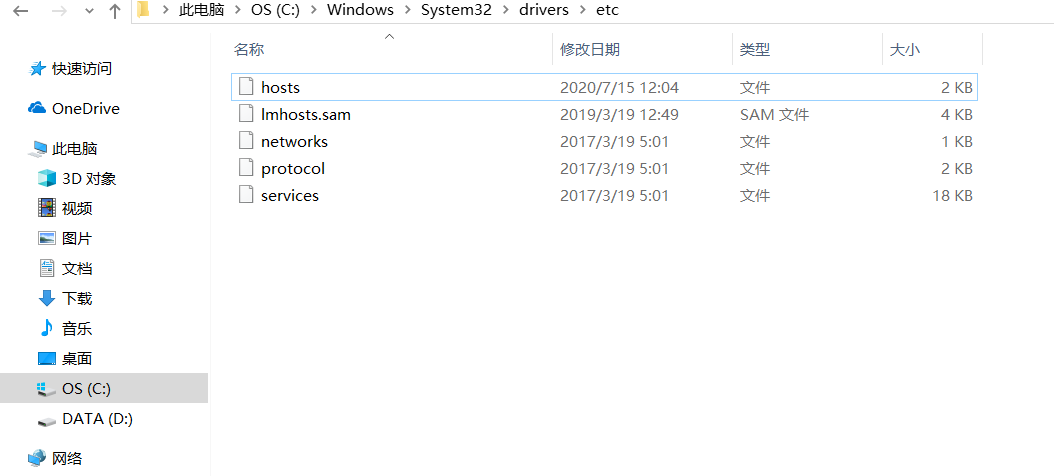
打开index.html后，将其修改成如下所示

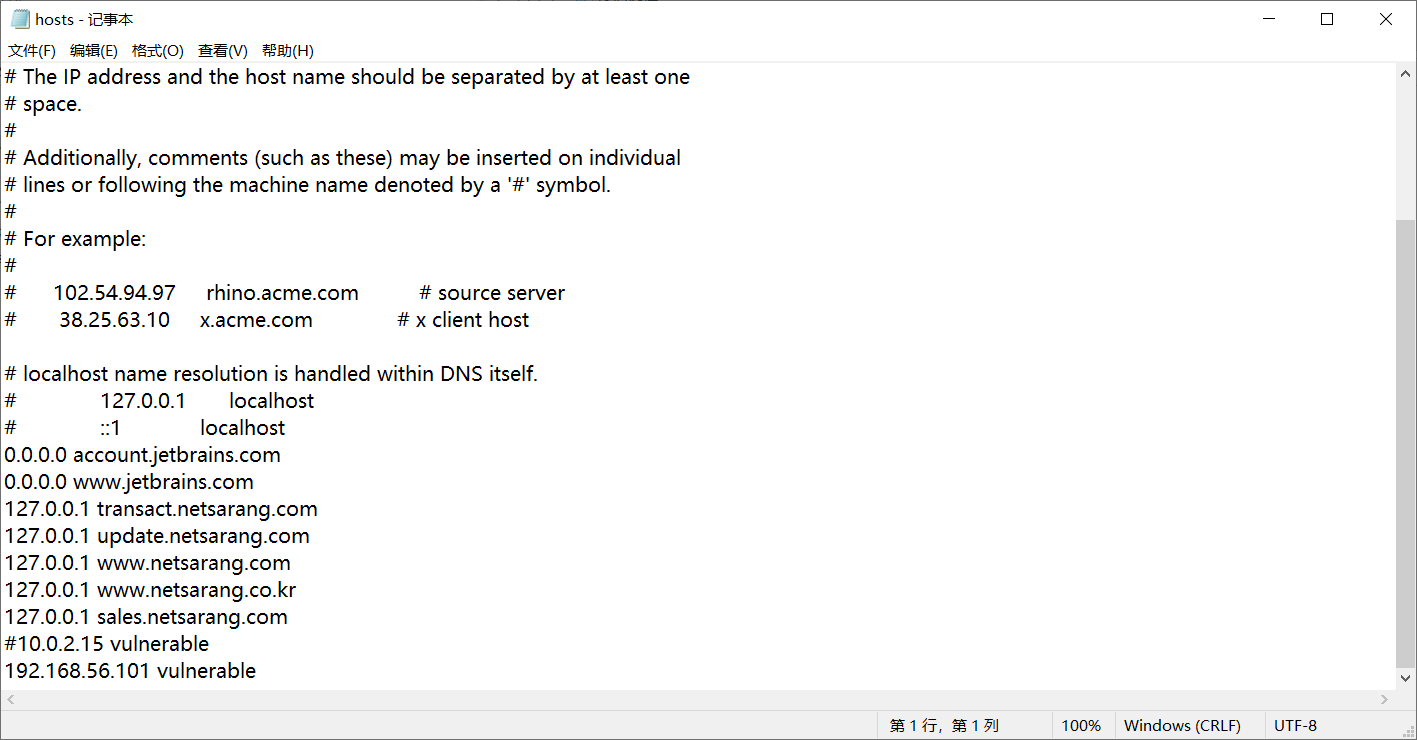
原网页最终显示如下：

**任务二：通过host文件解析名称**

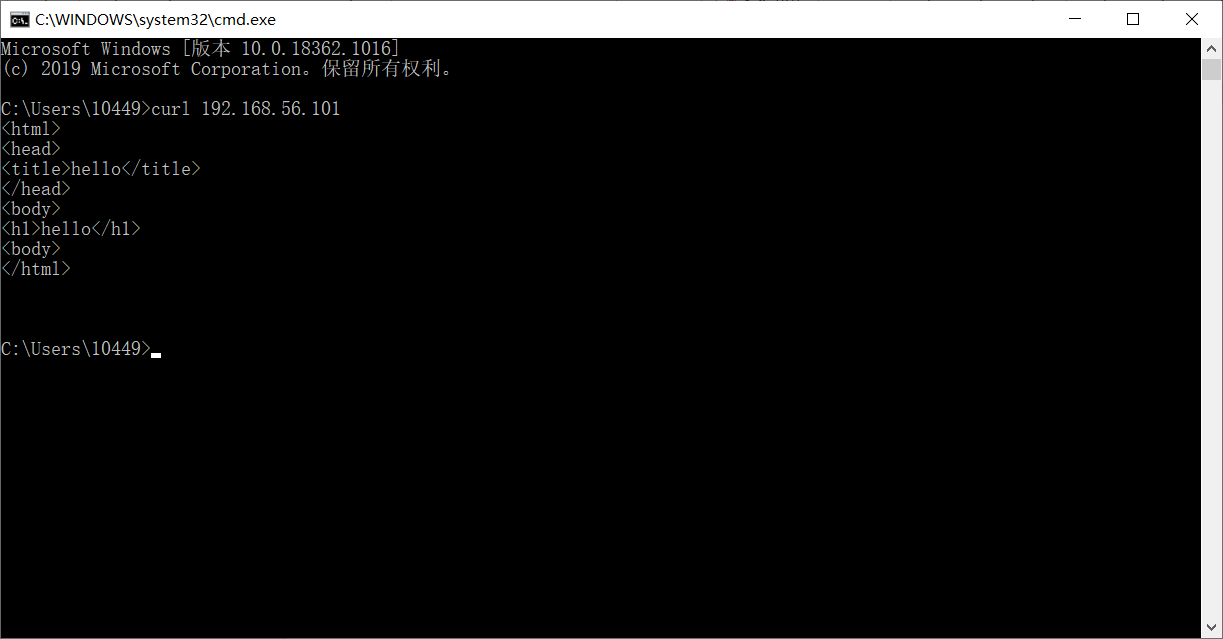
首先通过命令查询虚拟机的ip地址：

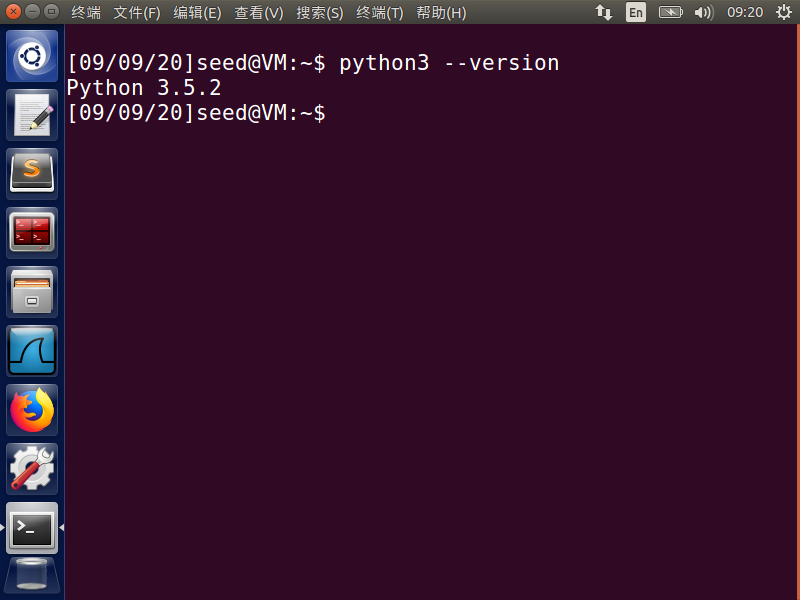
得到为192.168.56.101/24

在windows主机中找到hosts文件记事本打开：

加入虚拟机ip地址和主机名vulnerable并保存

**任务三：编写HTTP客户端，使用http库检索站点的主页**

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

查看虚拟机python版本：

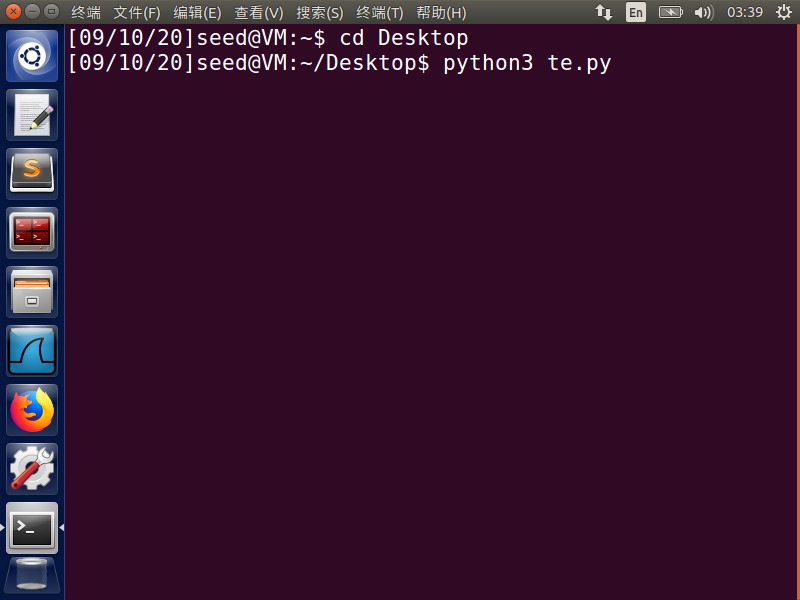
将以下代码保存为te.py：

import requests

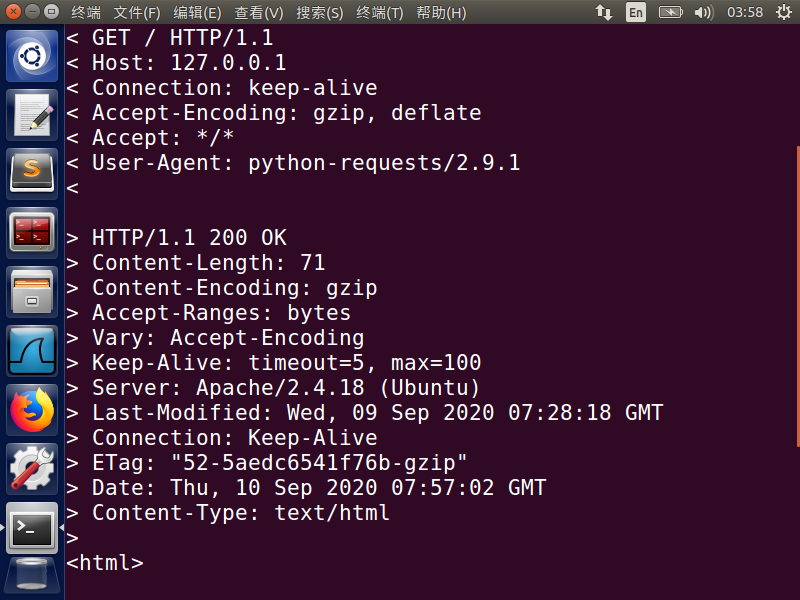
from requests\_toolbelt.utils import dump

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

data =dump.dump\_all(resp)

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

执行te.py，结果如下



**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**

#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() {

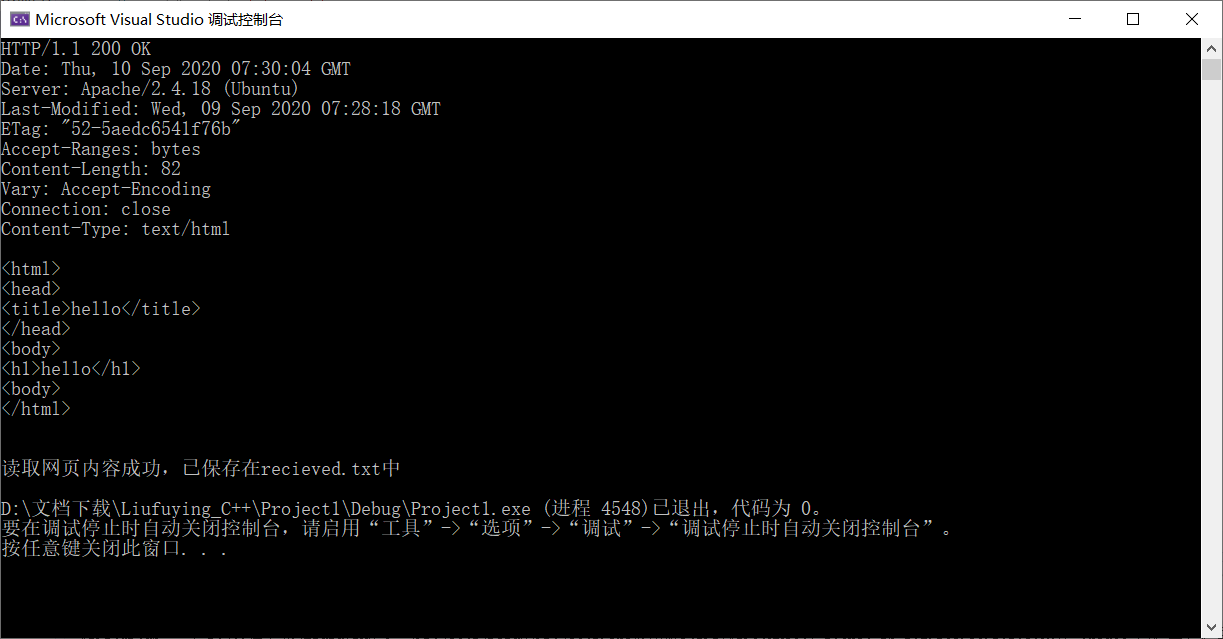
const char\* str = "vulnerable";

ReadPage(str);

return 0;

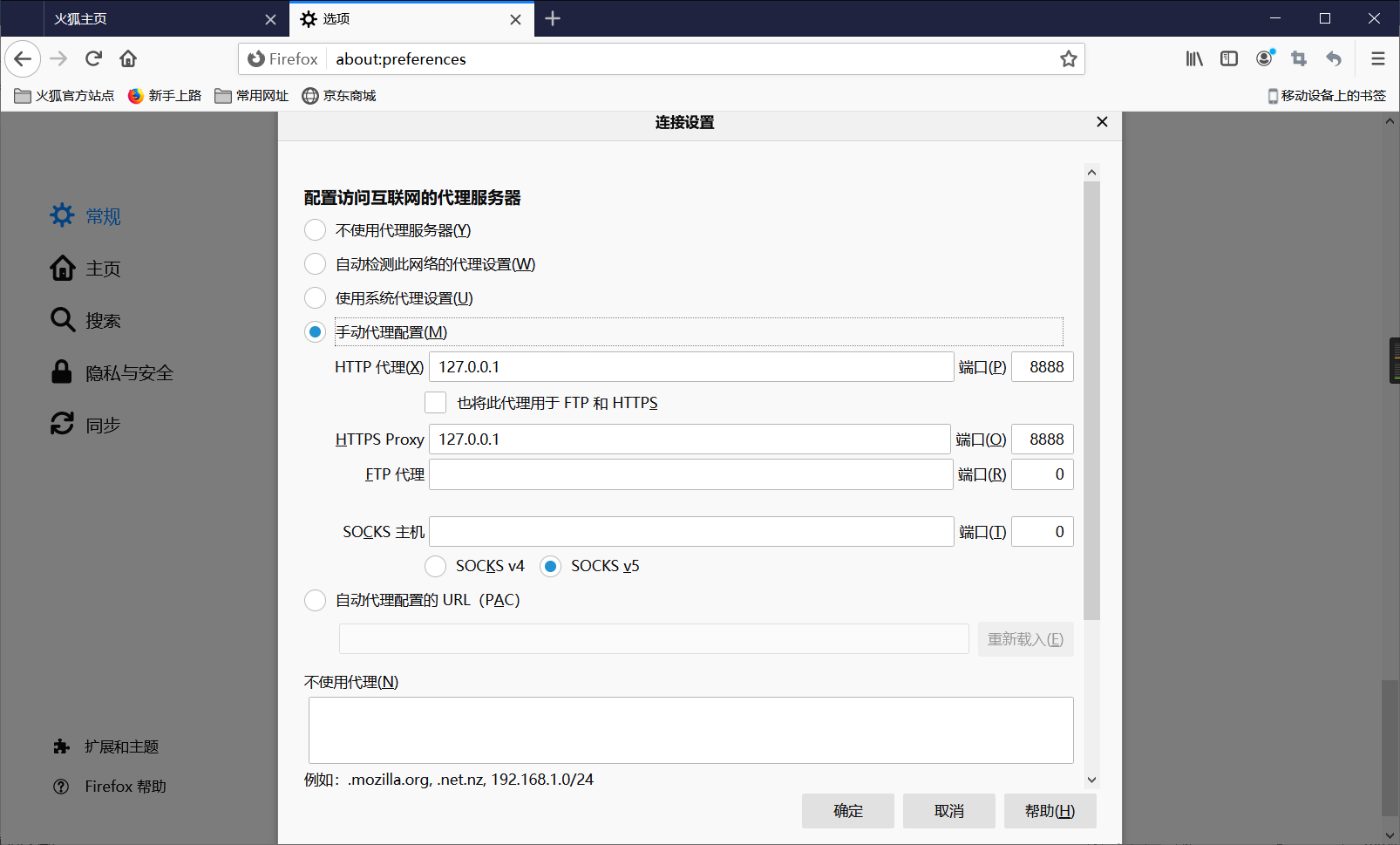
system("pause");

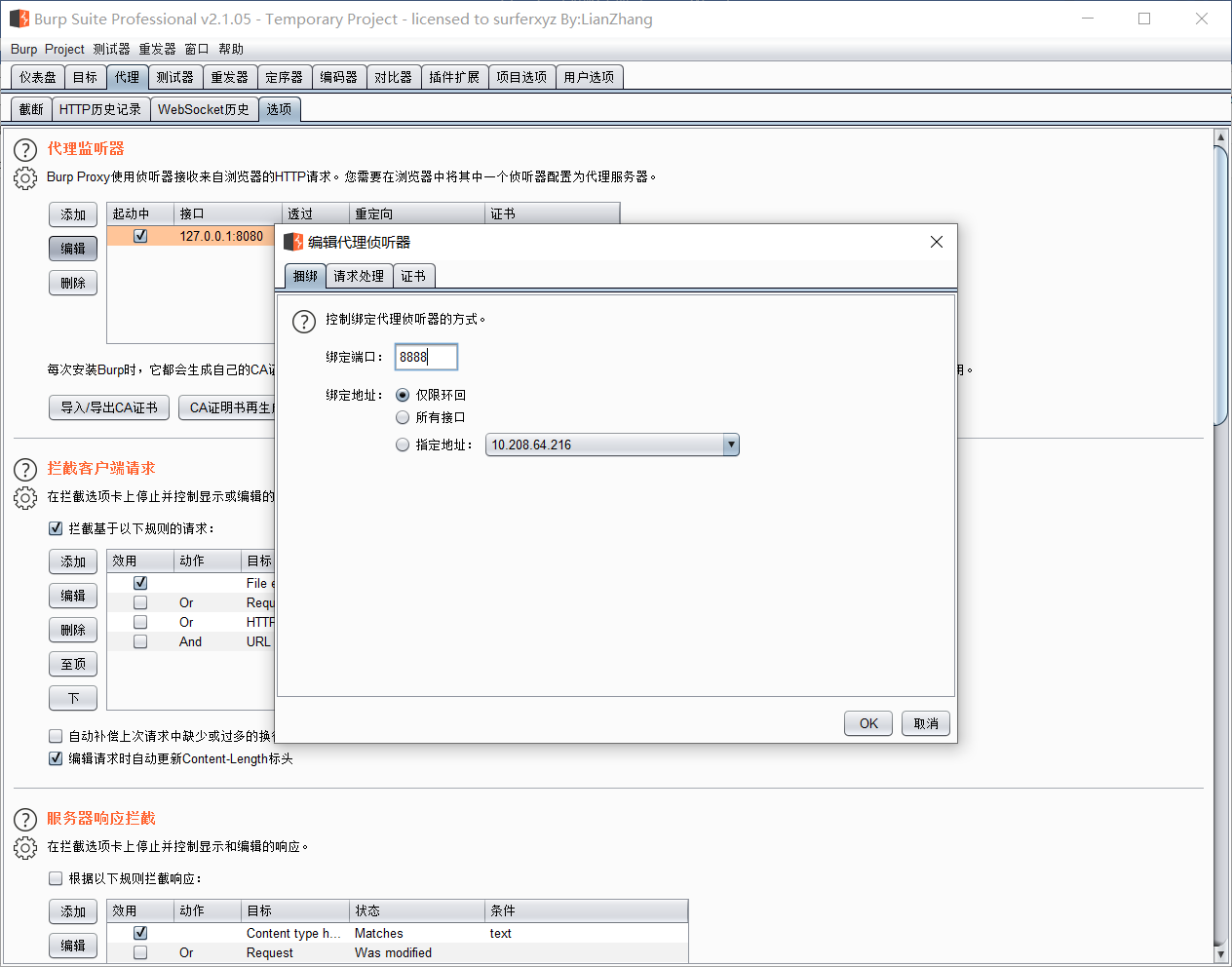
}

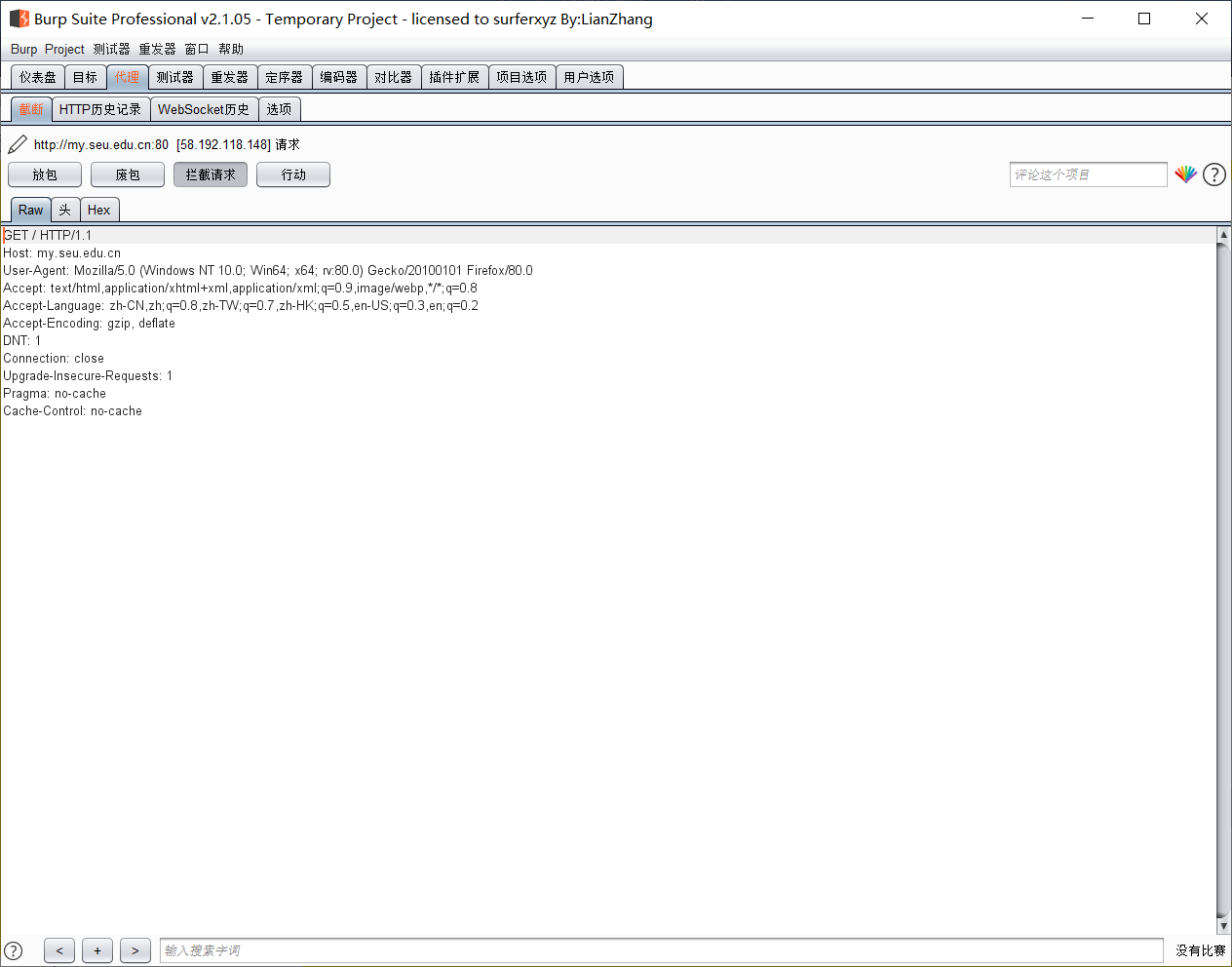
执行结果如下：

**任务五：下载软件Burp Suite并访问网站查看请求与响应的信息**

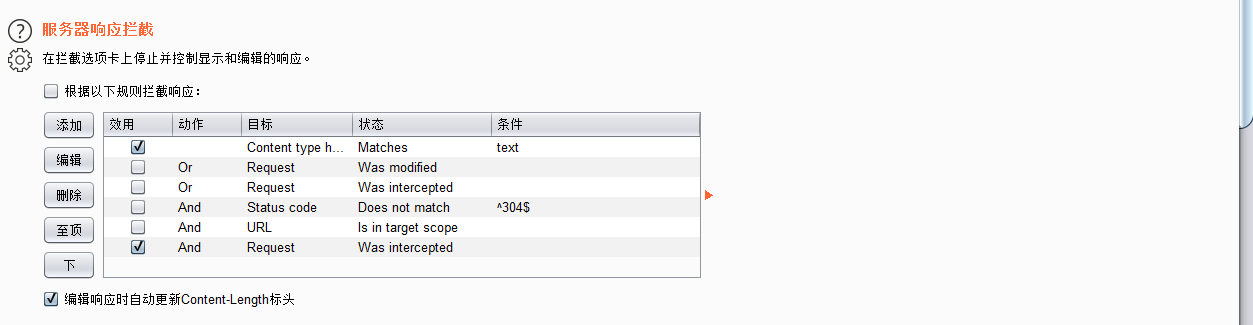
因chrome版本问题于是选用Firefox进行实验

设置代理，地址设为127.0.0.1,端口修改为8888

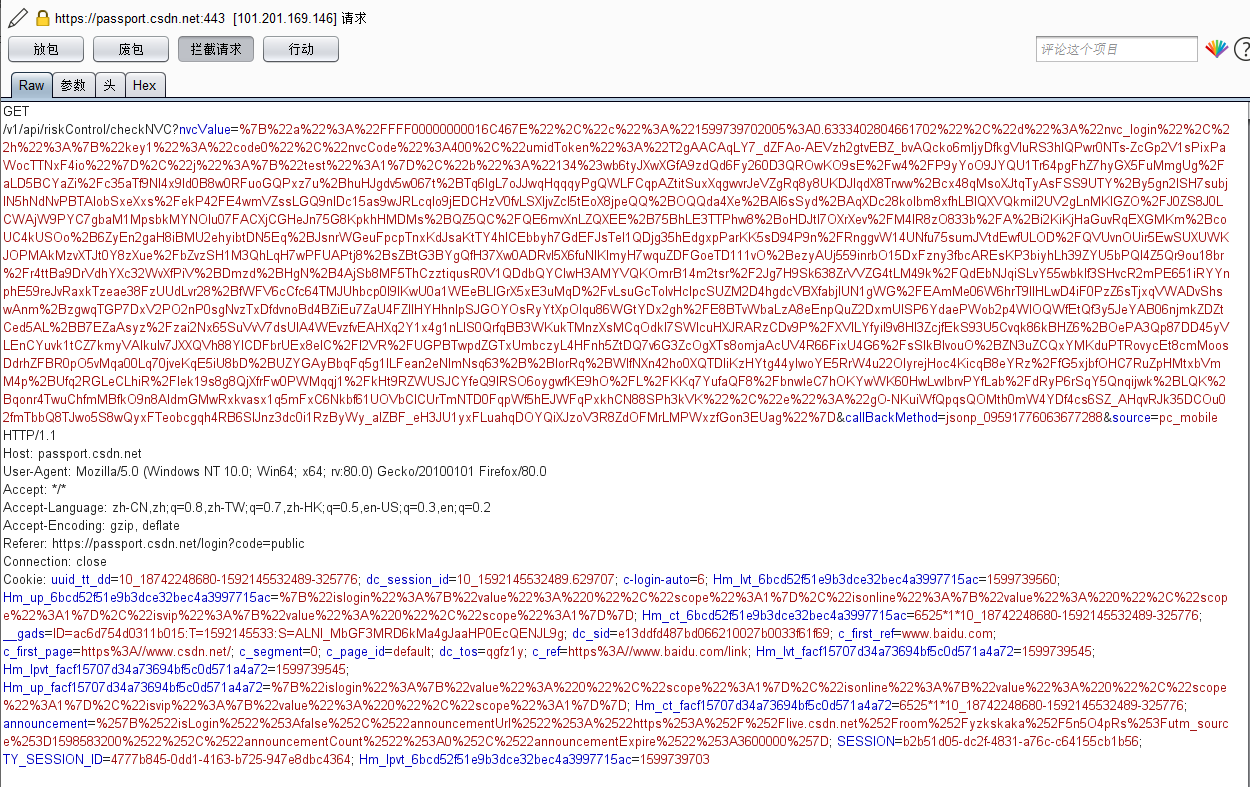
打开Burp Suite界面，设置Proxy代理，端口改为8888

使用浏览器打开my.seu.edu.cn查看拦截情况

更改服务器响应拦截设置



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

请求：

响应：

