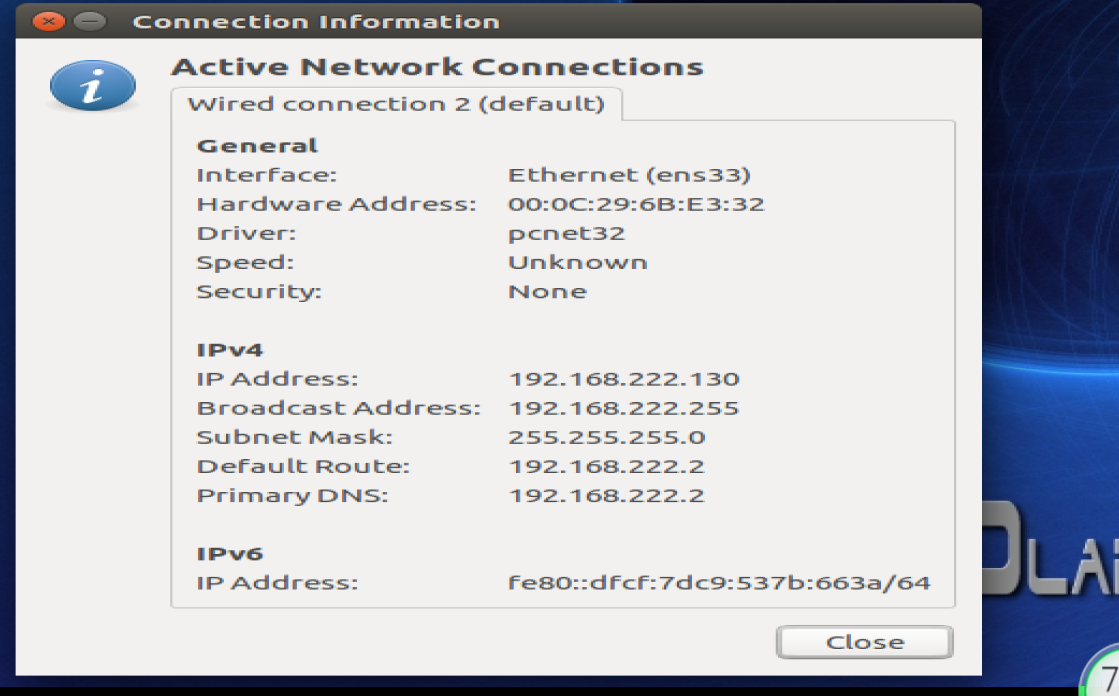
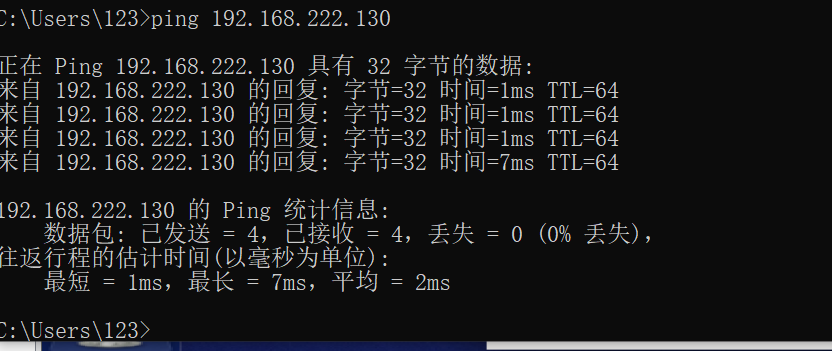
查看虚拟机ip地址

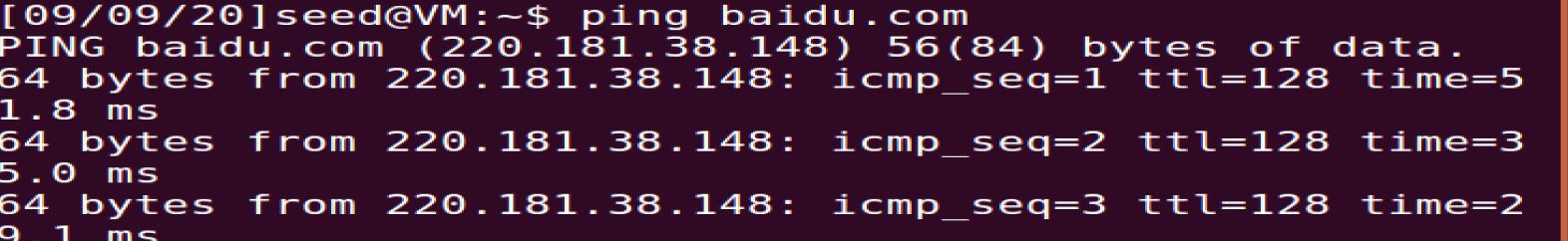


发现虚拟机ip地址为192.168.222.130

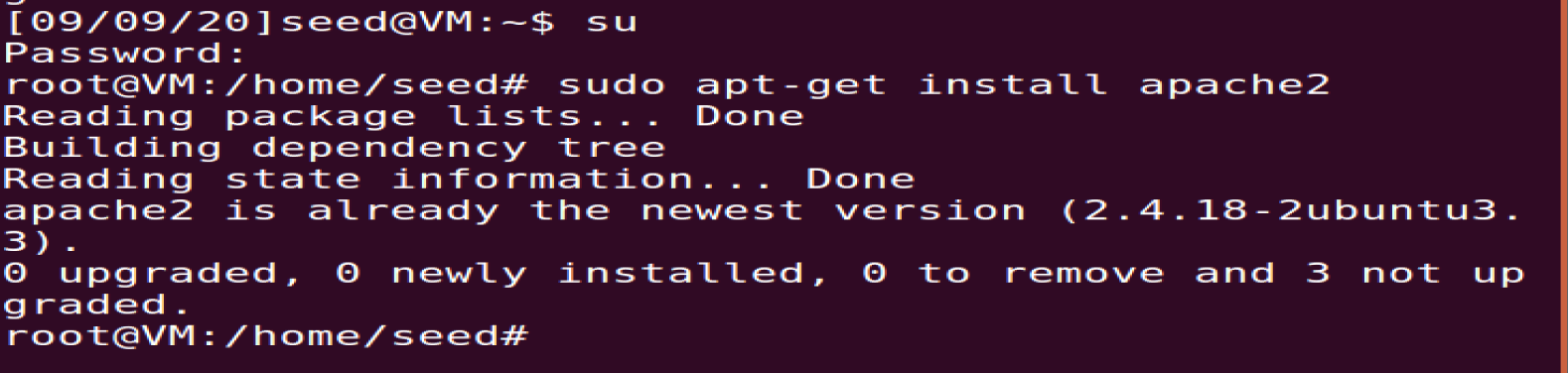
在主机上ping地址192.168.222.130发现能够行的通



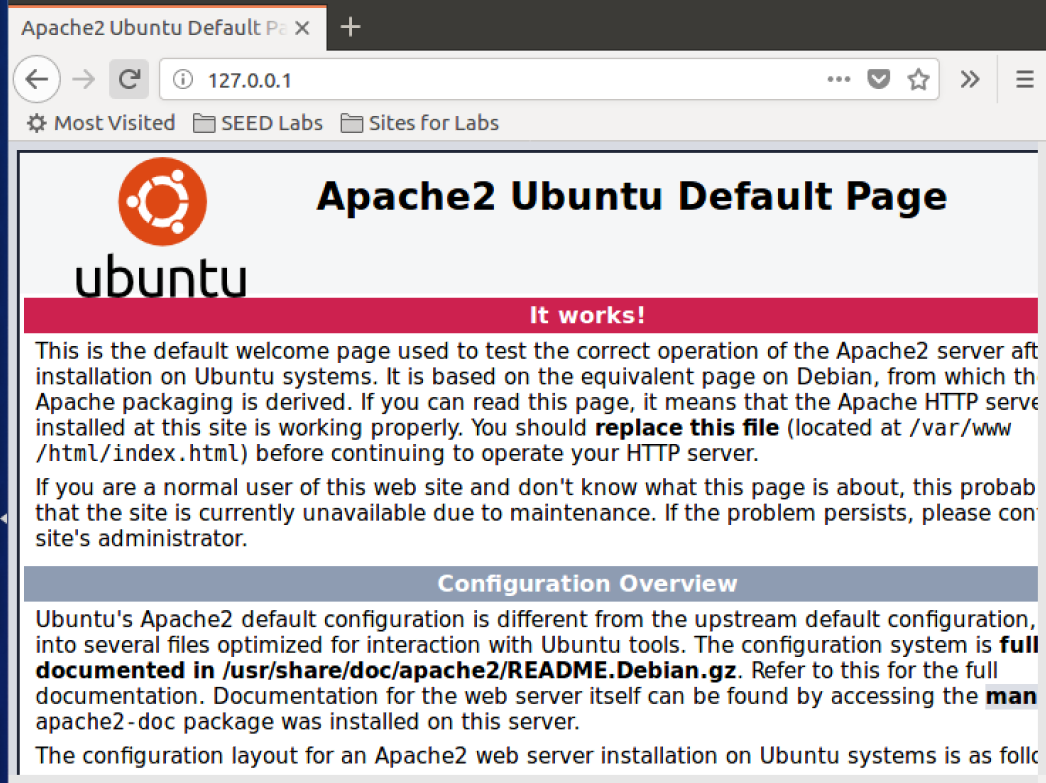
虚拟机ping百度发现能行，表示虚拟机已经联网



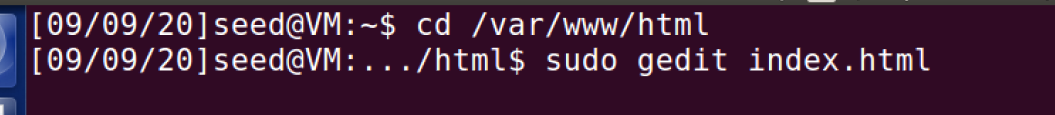
安装apache

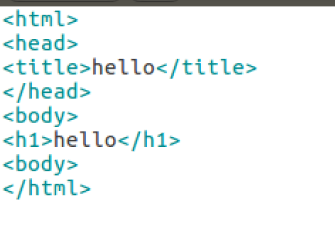


打开127.0.0.1网站

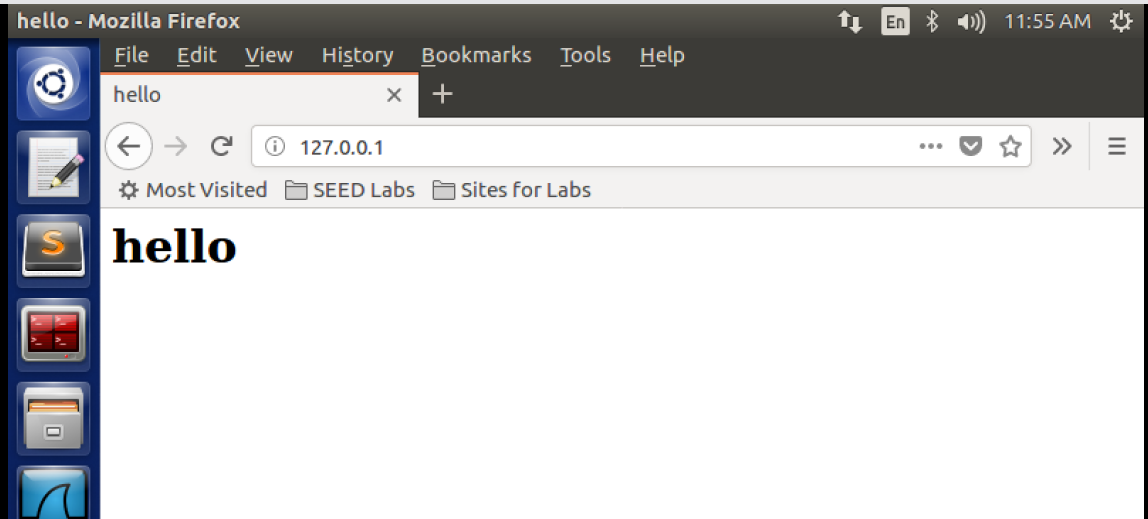


修改127.0.0.1网站

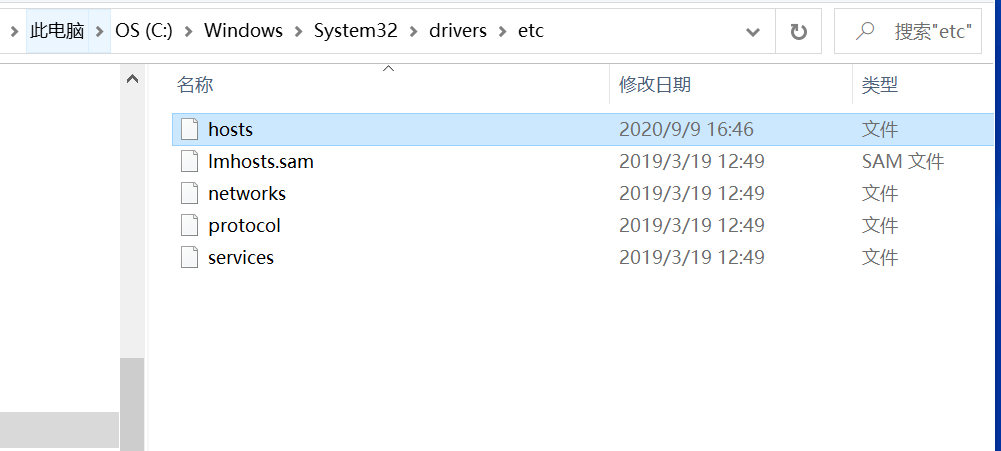




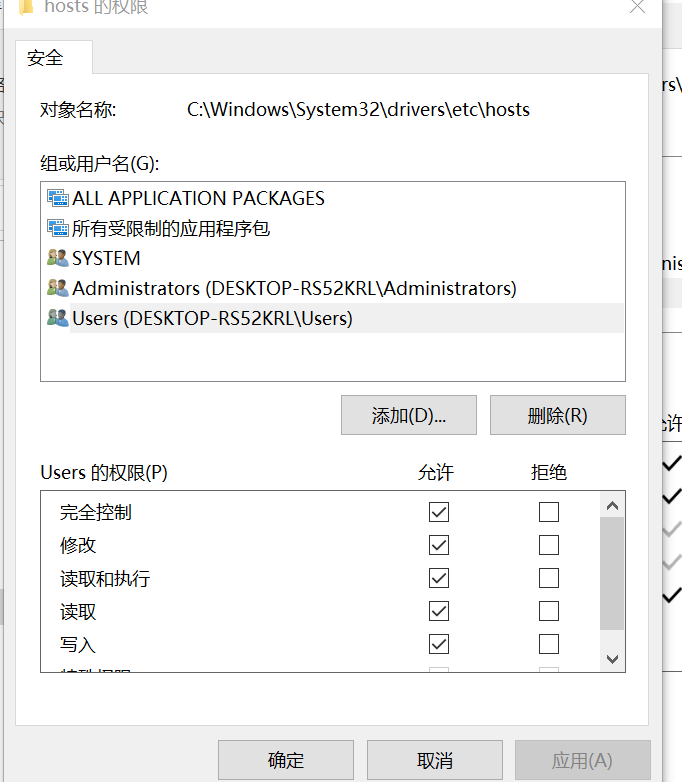
重新打开127.0.0.1网站



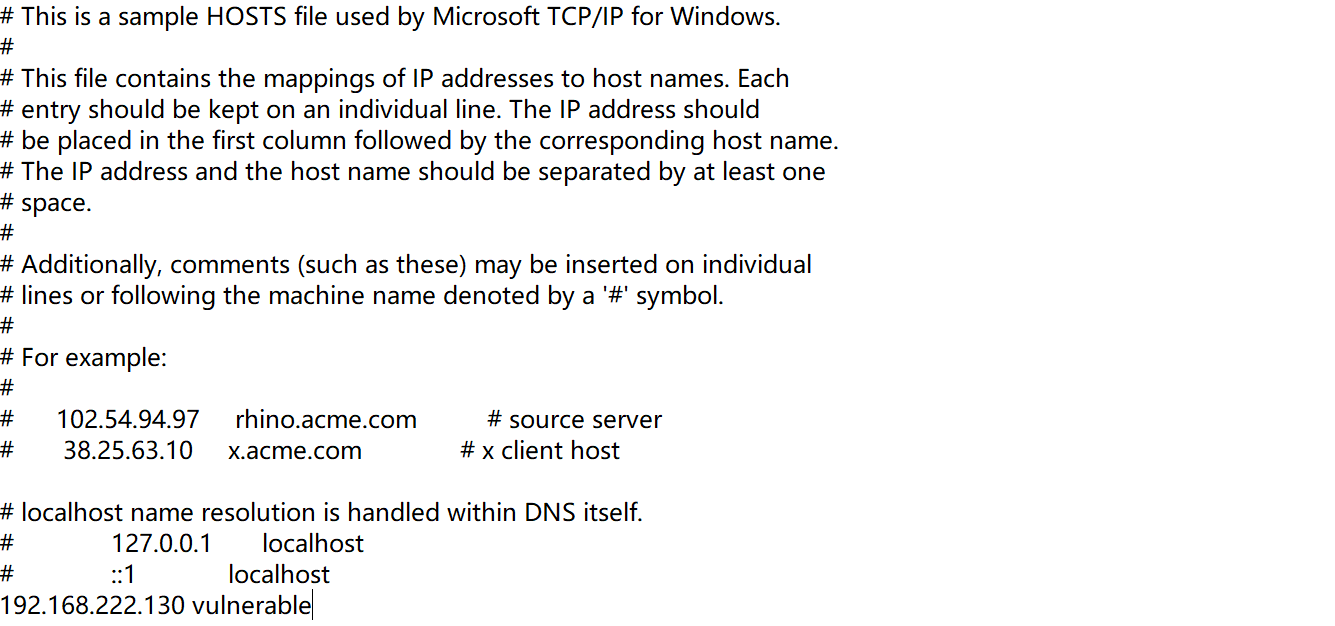
**通过host文件解析名称**



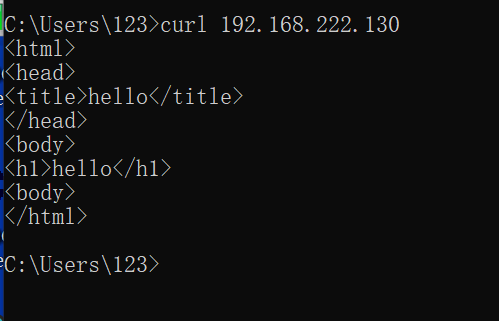
修改权限，使得拥有修改hosts权限



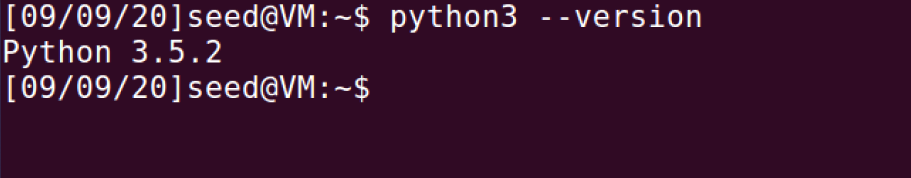
修改hosts



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



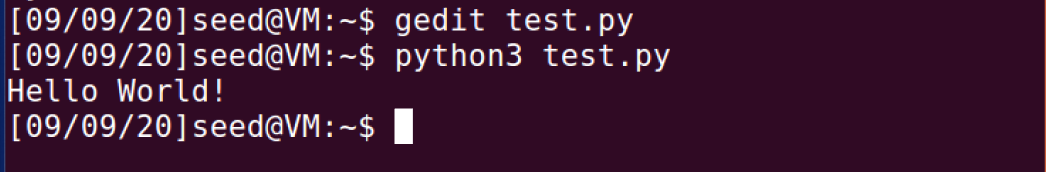
虚拟机中输入python3 --version 查看虚拟机是否有python3.5



创建.py的python执行文件（新建document，用gedit进行编辑，编辑结束重命名为xxx.py）

命令行转到文件所在位置（可以右键properties查看），输入命令

python3 test.py 得到信息如下



**编写HTTP客户端以使用套接字检索站点的主页，代码如下：**

在vc6加一句#pragma comment(linker, "/entry:mainCRTStartup")编译

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

#include <winsock2.h>

#include<time.h>

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

#pragma comment(linker, "/entry:mainCRTStartup")

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() {

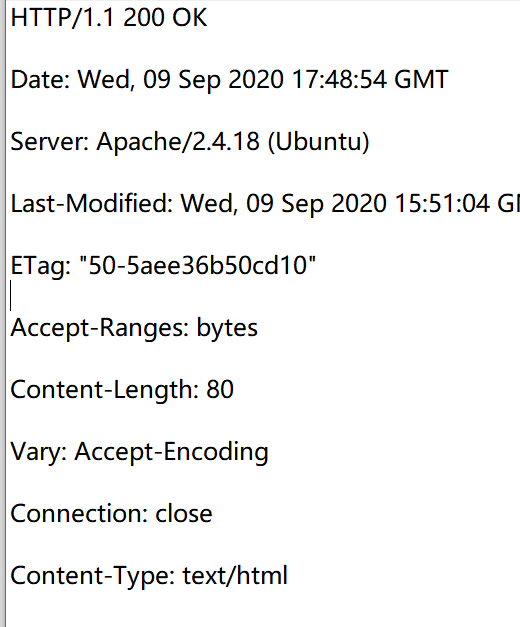
char\* str = "vulnerable";

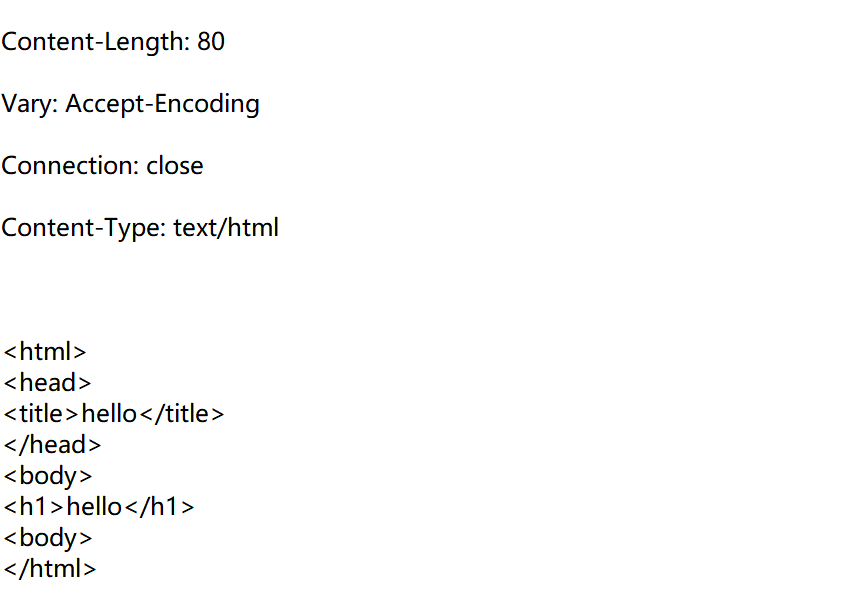
ReadPage(str);

system("pause");

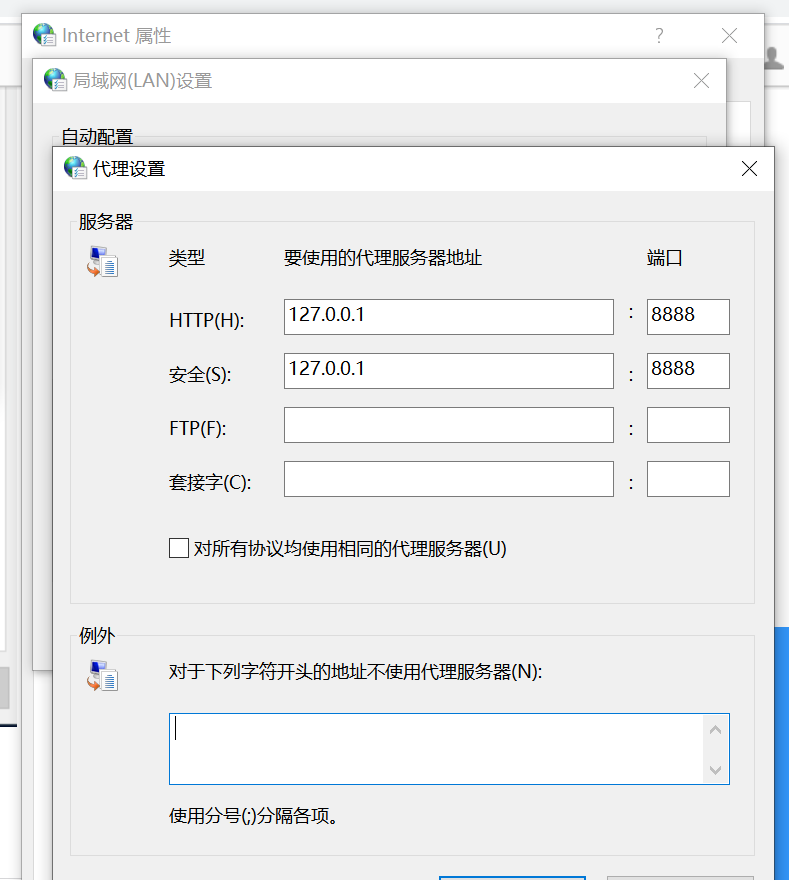
}

最后得到

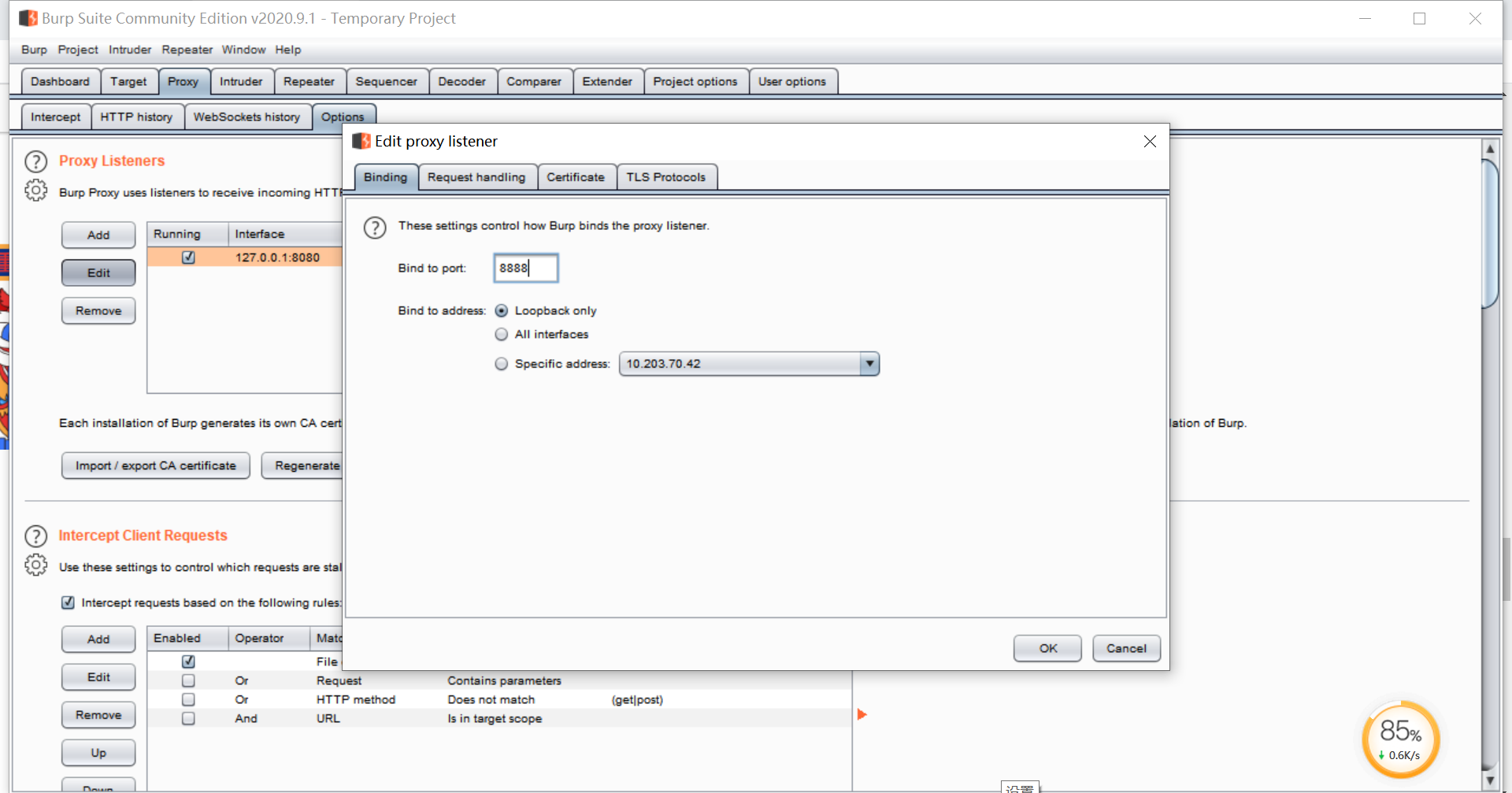




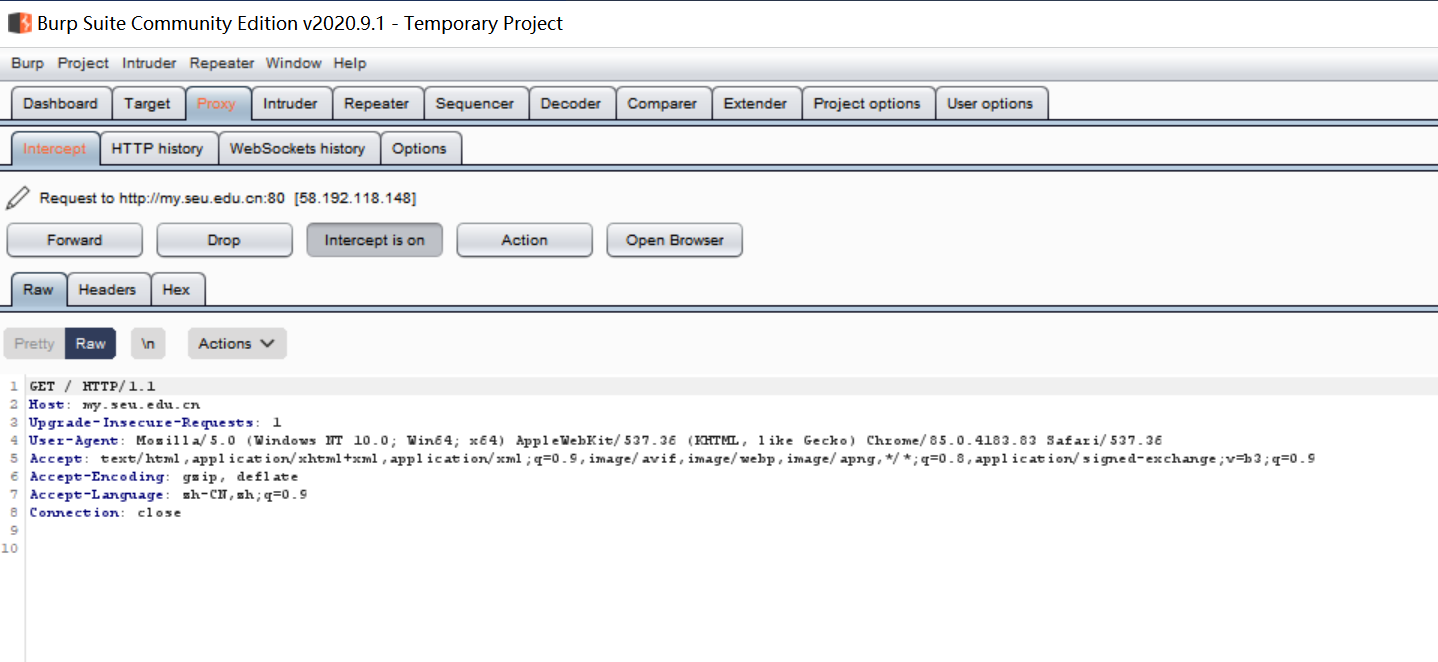
对测试浏览器Chrome进行代理设置,地址设为127.0.0.1,端口修改为8888

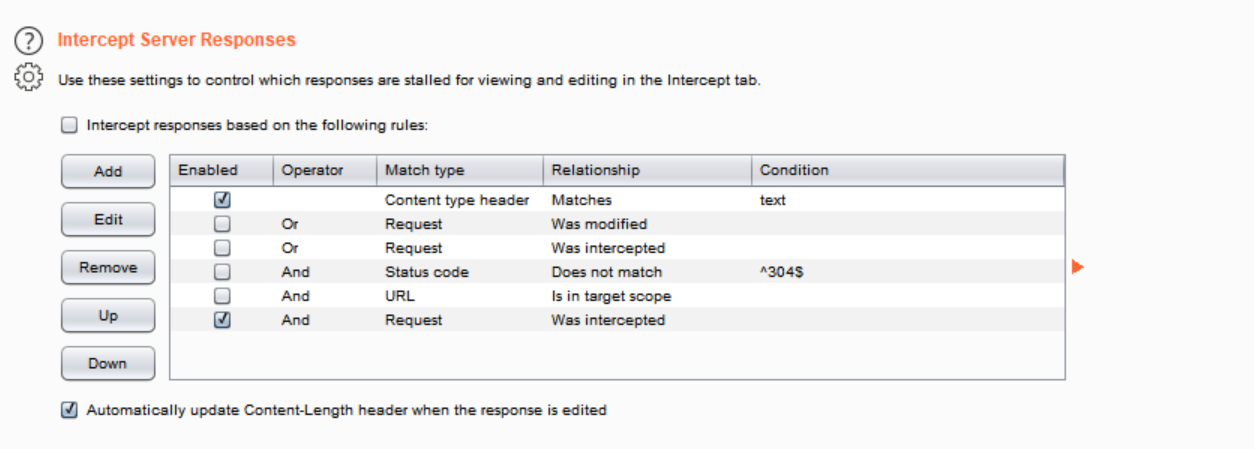


打开Burp Suite界面，设置Proxy代理，端口改为8888（需安装CA安全证书）



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）





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)

