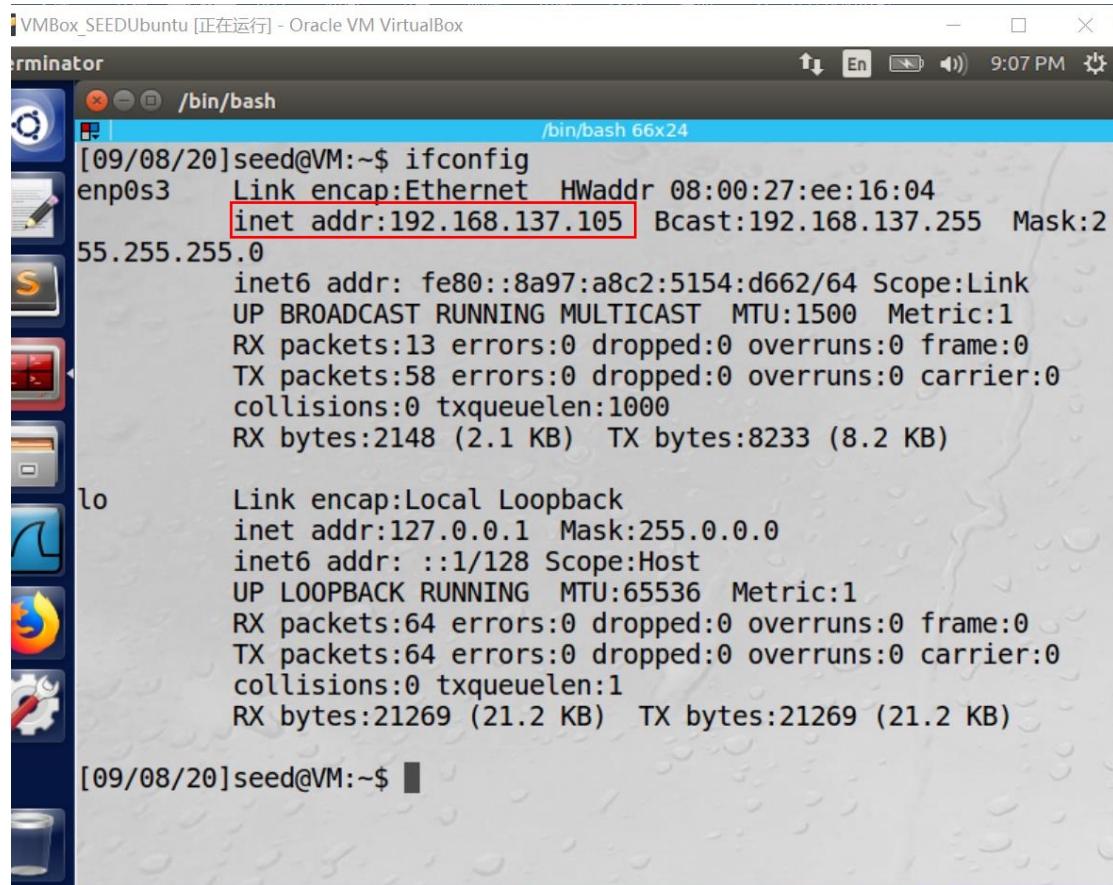


HTTP 基础实验报告

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：

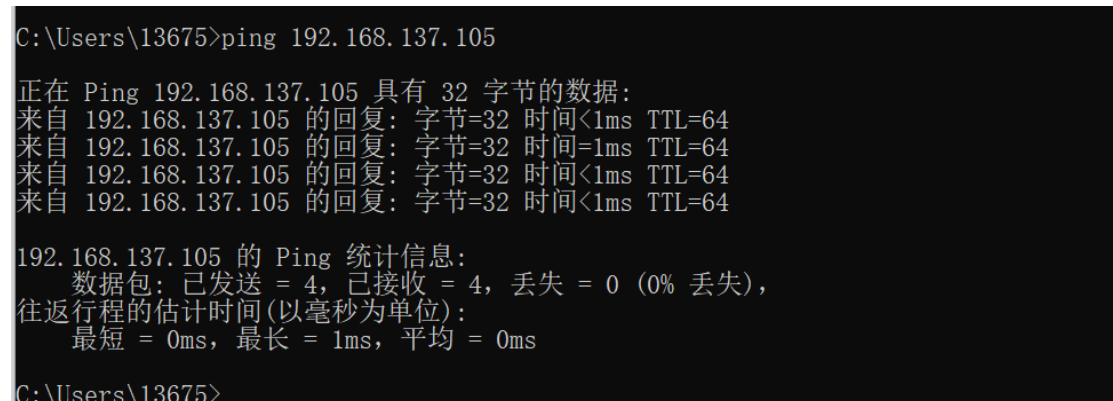


```
[09/08/20]seed@VM:~$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:ee:16:04
            inet addr:192.168.137.105  Bcast:192.168.137.255  Mask:255.255.255.0
            inet6 addr: fe80::8a97:a8c2:5154:d662/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:13 errors:0 dropped:0 overruns:0 frame:0
            TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2148 (2.1 KB)  TX bytes:8233 (8.2 KB)

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:64 errors:0 dropped:0 overruns:0 frame:0
            TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:21269 (21.2 KB)  TX bytes:21269 (21.2 KB)

[09/08/20]seed@VM:~$
```

在主机上 ping 虚拟机，成功：



```
C:\Users\13675>ping 192.168.137.105

正在 Ping 192.168.137.105 具有 32 字节的数据:
来自 192.168.137.105 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.137.105 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.137.105 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.137.105 的回复: 字节=32 时间<1ms TTL=64

192.168.137.105 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Users\13675>
```

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-get install apache`，安装 apache 服务器：

```
[09/08/20]seed@VM:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.18-2ubuntu3.3).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

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：

```
[09/08/20]seed@VM:~$ cd /var/www/html
[09/08/20]seed@VM:.../html$ ls
index.html
```



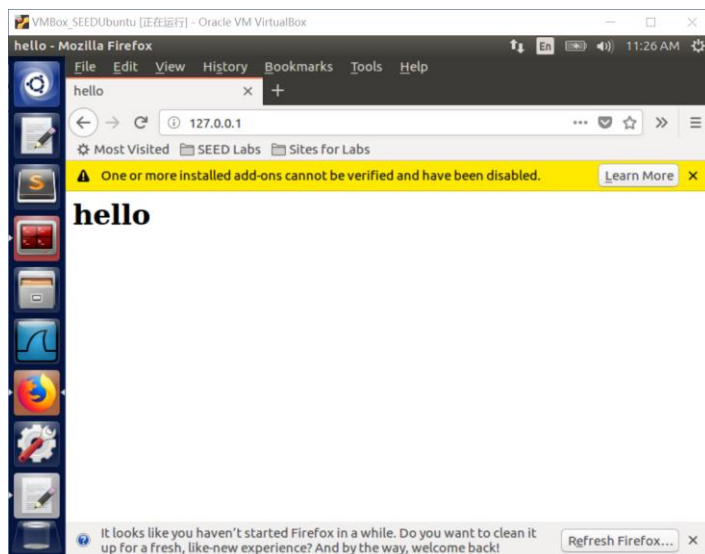
3、使用 `sudo gedit index.html` 指令打开 `index.html` 文件并进行编写：

```
[09/08/20]seed@VM:.../html$ sudo gedit index.html
```

内容如下：

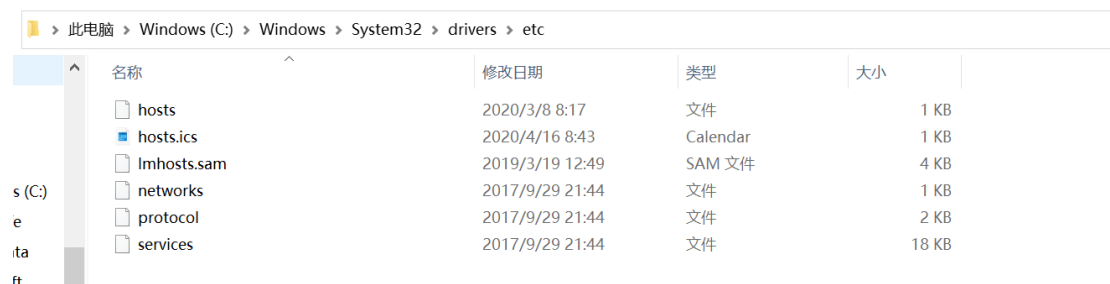
```
<html>
<head>
<title>hello</title>
</head>
<body>
<h1>hello</h1>
<body>
</html>
```

4、修改后使用浏览器登录 127.0.0.1，页面更新为新主页：

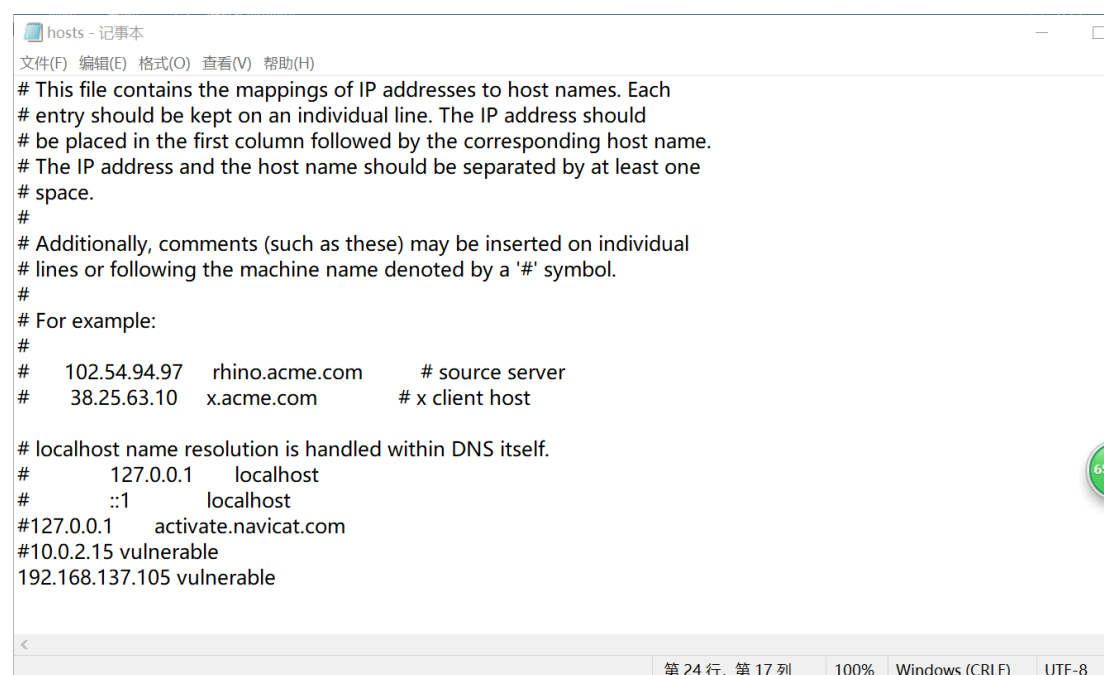


Task2:通过 host 文件解析名称

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：

```
C:\Users\13675>curl 192.168.137.105
<html>
<head>
<title>hello</title>
</head>
<body>
<h1>hello</h1>
</body>
</html>
C:\Users\13675>
```

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

```
[09/08/20]seed@VM:~$ python3 --version
Python 3.5.2
[09/08/20]seed@VM:~$ █
```

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 = WSStartup(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
memcpy(&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：

 D:\课程资料和作业\网安实训\课程实验\实验三任务四\Debug\实验三任务四.exe

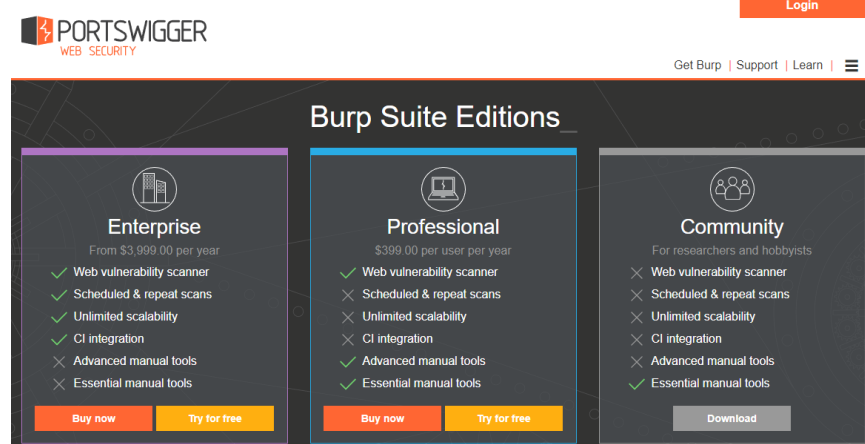
```
HTTP/1.1 200 OK  
Date: Wed, 09 Sep 2020 02:00:26 GMT  
Server: Apache/2.4.18 (Ubuntu)  
Last-Modified: Tue, 08 Sep 2020 15:24:37 GMT  
ETag: "51-5aeceeed57537"  
Accept-Ranges: bytes  
Content-Length: 81  
Vary: Accept-Encoding  
Connection: close  
Content-Type: text/html
```

```
<html>  
<head>  
<title>hello</title>  
</head>  
<body>  
<h1>hello</h1>  
</body>  
</html>
```

读取网页内容成功，已保存在recieved.txt中
请按任意键继续. . .

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 Community 版本：



The image shows the 'Burp Suite Editions' page from Portswigger. It features three columns for different editions: Enterprise, Professional, and Community. Each column lists features with checkmarks for included features and crosses for excluded ones. The Community edition is highlighted with a blue border and has a 'Download' button.

Edition	Price	Web vulnerability scanner	Scheduled & repeat scans	Unlimited scalability	CI integration	Advanced manual tools	Essential manual tools
Enterprise	From \$3,999.00 per year	✓	✓	✓	✓	✗	✗
Professional	\$399.00 per user per year	✓	✗	✗	✗	✓	✓
Community	For researchers and hobbyists	✗	✗	✗	✗	✗	✓

下载并安装 jdk，配置环境变量：

新建系统变量

变量名(N):

变量值(V):

新建系统变量

变量名(N):

变量值(V):

新建系统变量

变量名(N):

变量值(V):

配置成功：

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18363.1016]
(c) 2019 Microsoft Corporation。保留所有权利。

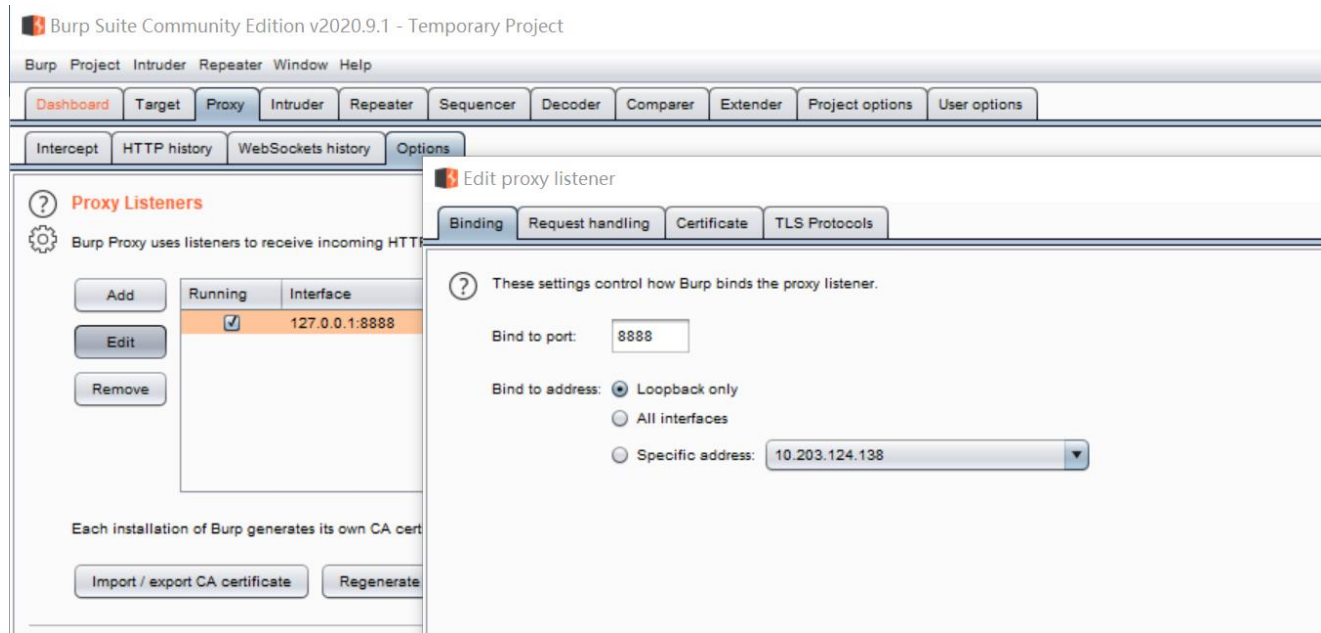
C:\Users\13675>java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

C:\Users\13675>
```

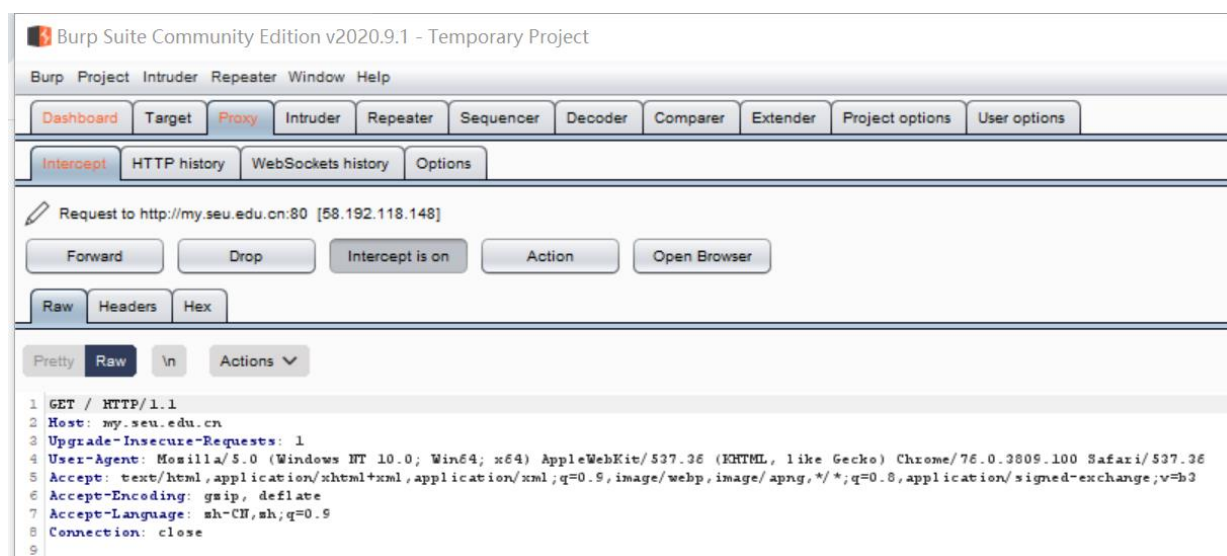
2、用 open browser 打开 Chrome 浏览器，并对浏览器进行代理设置，地址设为 127.0.0.1, 端口修改为 8888:



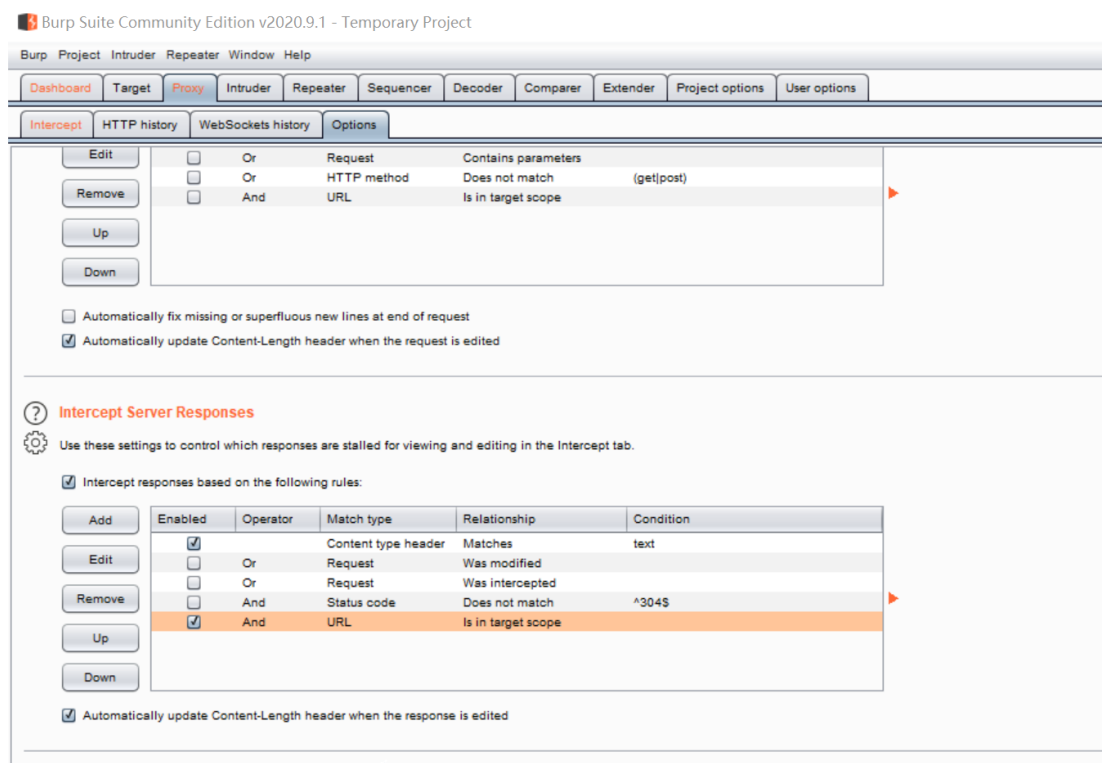
3、打开 Burp Suite 界面，设置 Proxy 代理，端口改为 8888:



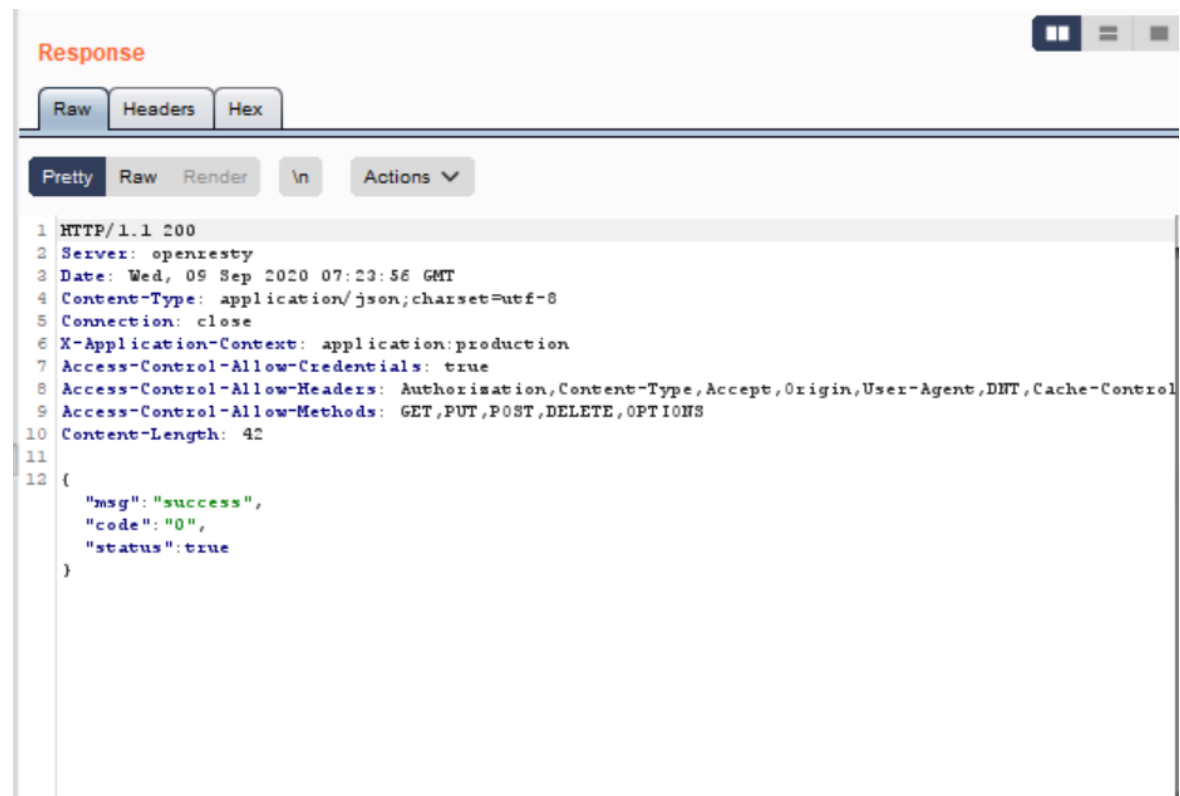
4、使用浏览器打开 my.seu.edu.cn 查看拦截情况：



5、配置拦截服务器响应的参数：



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



The screenshot shows the 'Response' tab in a web browser's developer tools. The response is an HTTP 200 status with various headers and a JSON body. The JSON body contains a success message, a code of 0, and a status of true.

```
1 HTTP/1.1 200
2 Server: openresty
3 Date: Wed, 09 Sep 2020 07:23:56 GMT
4 Content-Type: application/json; charset=utf-8
5 Connection: close
6 X-Application-Context: application:production
7 Access-Control-Allow-Credentials: true
8 Access-Control-Allow-Headers: Authorisation, Content-Type, Accept, Origin, User-Agent, DNT, Cache-Control
9 Access-Control-Allow-Methods: GET, PUT, POST, DELETE, OPTIONS
10 Content-Length: 42
11
12 {
  "msg": "success",
  "code": "0",
  "status": true
}
```

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。