

Lab3 实验

实验目的：

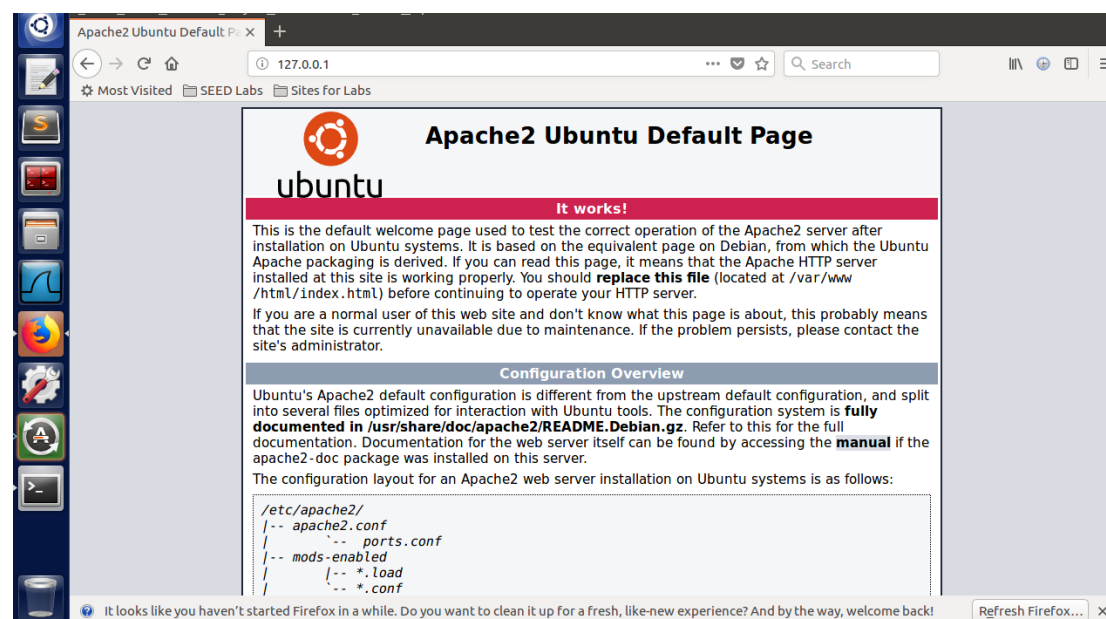
通过该次实验，我们对 http 网页的基本操作熟练掌握，并且可以获得对 Burp Suite 软件的基础操作，对拦截的信息可以进行简单分析，查看拦截文件的信息。

任务一：HTTP 基础

首先我们根据实验手册，在虚拟机中打开 terminal 终端窗口，输入 `sudo apt-get install apache2`

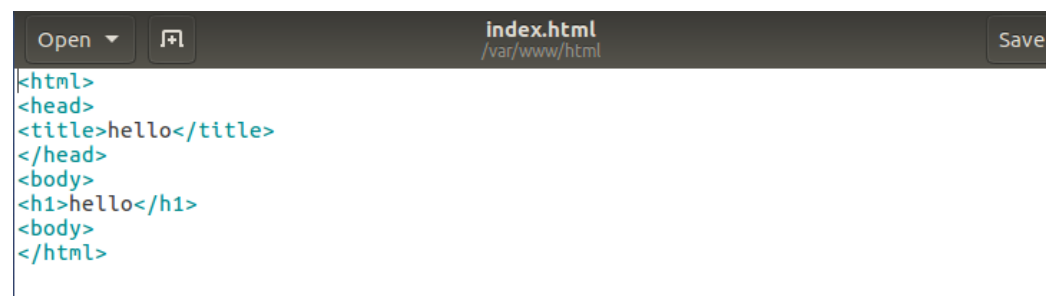
```
[09/08/20]seed@VM:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.18-2ubuntu3.3).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

接下来在虚拟机中输入“127.0.0.1”，打开网页界面。

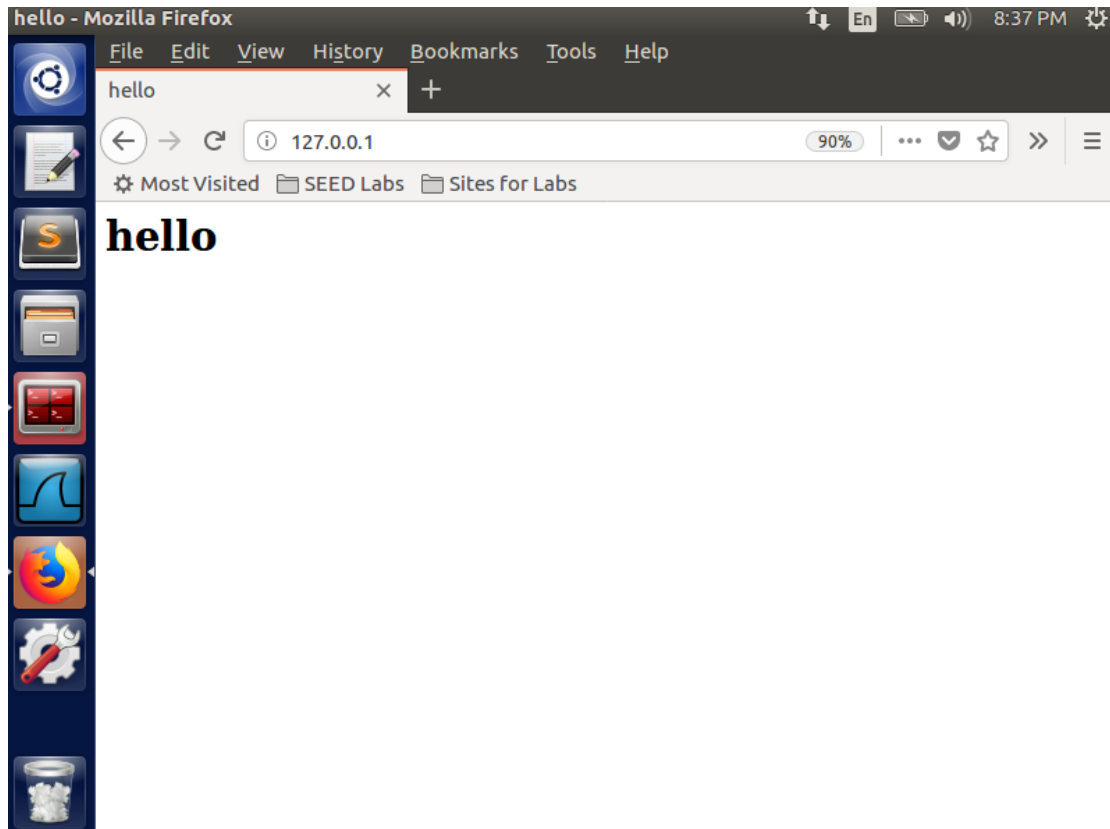


接下来依旧按照步骤，将使用 `sudo gedit index.html` 指令打开 `index.html` 并进行编写
按照实验手册的代码对

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



按照上图修改后再继续登录 127.0.0.1，页面更改为新主页



任务二：通过 host 文件解析名称

步骤一：在 windows 主机中找到 hosts 文件用记事本打开，然后修改 hosts 文件的虚拟 IP 地址与主机名（保存要以管理员身份运行记事本）。虚拟机的 ip 地址，根据实验最开始的环境配置，我们将虚拟机设置为桥接模式，然后在虚拟机上获取 IP 地址 192.168.137.155，之后我们再用主机 ping 虚拟机，可以 ping 通。

```
hosts - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
#127.0.0.1 activate.navicat.com
#10.0.2.15 vnlnerable
192.168.137.155 vulnerable|
```

```
/bin/bash 66x24
[09/08/20]seed@VM:~$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:14:72:76
            inet addr:192.168.137.155  Bcast:192.168.137.255  Mask:255.255.255.0
            inet6 addr: fe80::a9ee:5b62:ca08:c62a/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:27 errors:0 dropped:0 overruns:0 frame:0
            TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:3484 (3.4 KB)  TX bytes:8939 (8.9 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:65 errors:0 dropped:0 overruns:0 frame:0
            TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:21320 (21.3 KB)  TX bytes:21320 (21.3 KB)
```

任务三：编写 HTTP 客户端，使用 http 库检索站点的主页

步骤一：Windows 主机中输入 curl 和 IP 地址，可查看编写的 Index 的文件

```
Microsoft Windows [版本 10.0.17763.1282]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\王雯>curl 192.168.137.155
<html>
<head>
<title>hello</title>
</head>
<body>
<h1>hello</h1>
<body>
</html>
```

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 test.py），之后命令行转到文件所在位置

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 = 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);

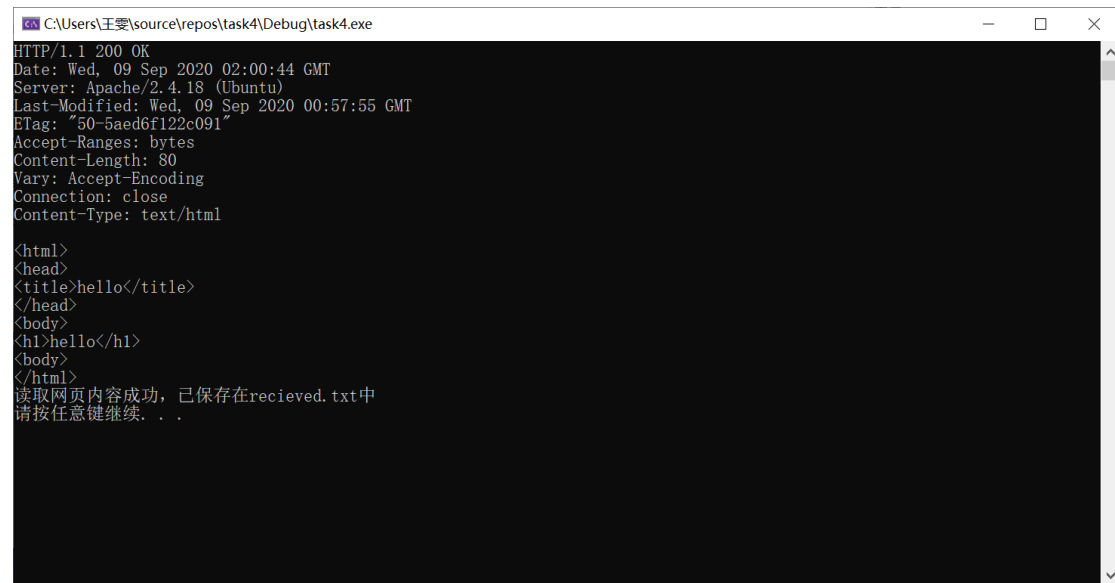
//Socket封装
struct sockaddr_in si;
si.sin_family = AF_INET;
si.sin_port = htons(80);
si.sin_addr.S_un.S_addr = inet_addr(ipstr);
int sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
connect(sock, (SOCKADDR*)&si, sizeof(si));
if (sock == -1 || sock == -2)
    return;

//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;
    }
}
```

然后执行该文件



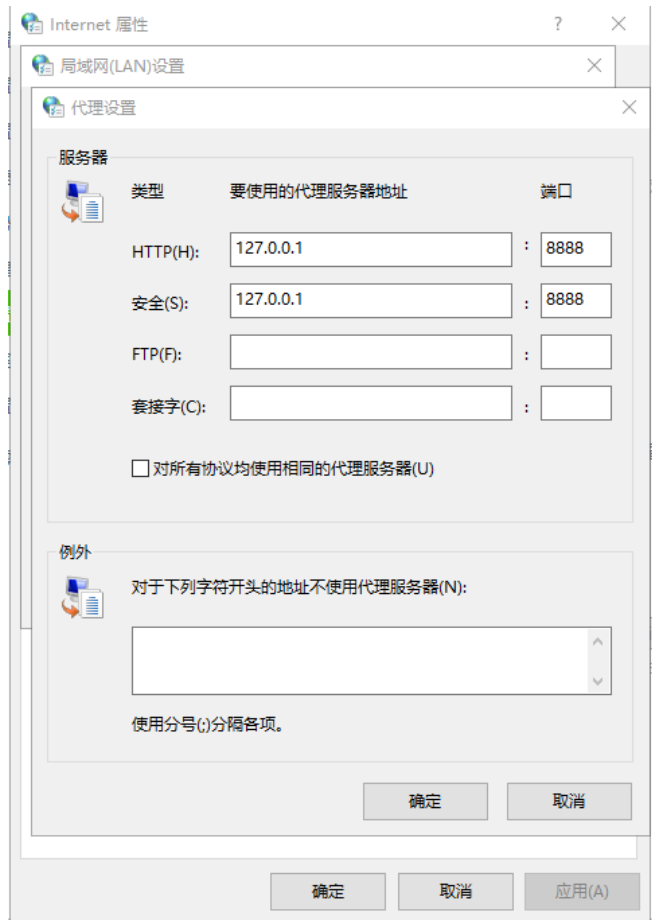
```
C:\Users\王雯\source\repos\task4\Debug\task4.exe
HTTP/1.1 200 OK
Date: Wed, 09 Sep 2020 02:00:44 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Wed, 09 Sep 2020 00:57:55 GMT
ETag: "50-5aed6f122c091"
Accept-Ranges: bytes
Content-Length: 80
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<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...
```

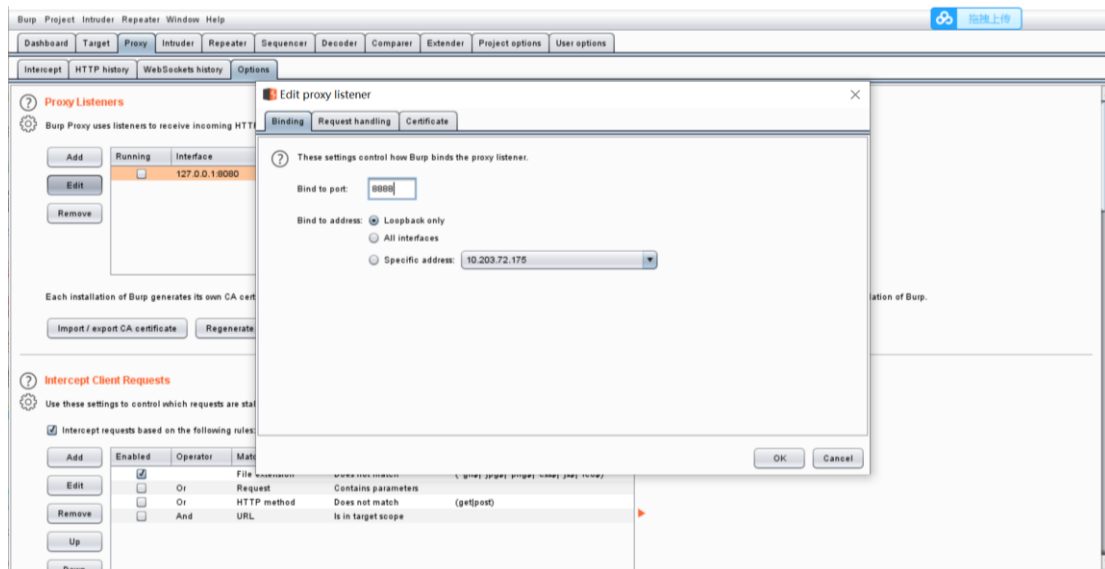
任务五：下载软件 **Burp Suite** 并访问网站查看请求与响应的信息

步骤一：先下载 BurpSuite

步骤二：对测试浏览器进行代理，地址设为 127.0.0.1，端口修改为 8888



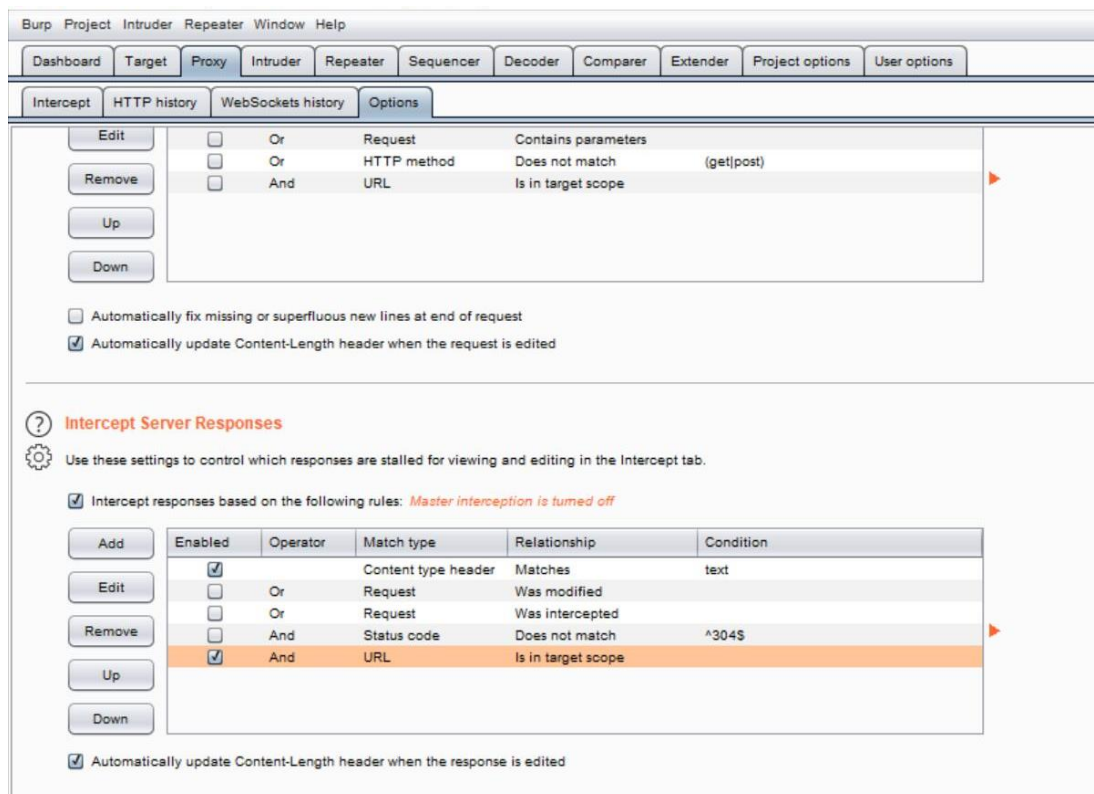
之后打开 Burp Suite 界面，设置 proxy 代理，端口改为 8888



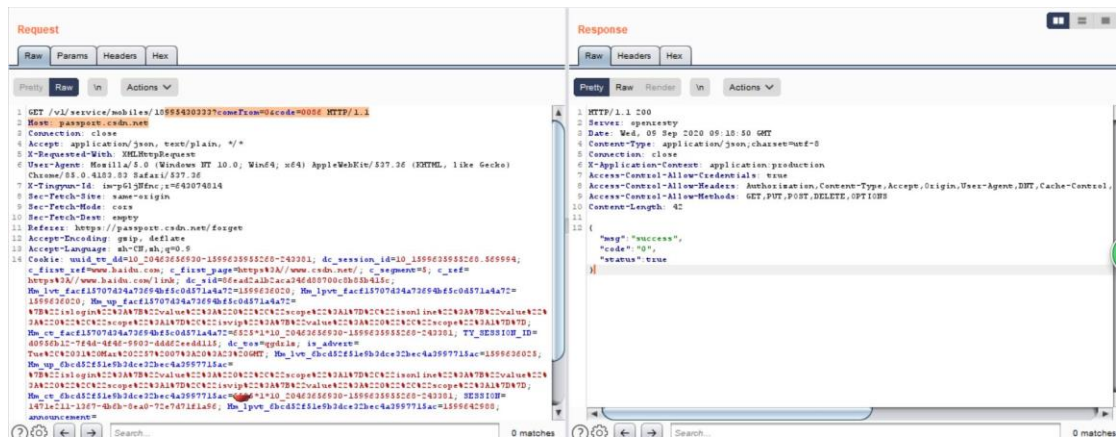
使用浏览器打开 my.seu.edu.cn 查看拦截情况



步骤五:



步骤六: 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)



实验总结：

在这次的实验中，我们对 http 的基础有了一定的掌握，同时可以通过终端修改代码对网页进行修改。同时我们通过下载 Burp Suite 软件，对基础的网络拦截，查看有了一定的了解。我们通过对 my.seu.edu.cn 还有 CSDN 的验证码功能的拦截，对 proxy 这个功能有了更深入的了解。实验中遇到的问题：在创建.py 文件后，无法通过终端命令的输入，得到与实验手册中的执行结果。在课后会继续请教助教，老师解决这个问题。