

实验 3 搭建简单的 Web 站点

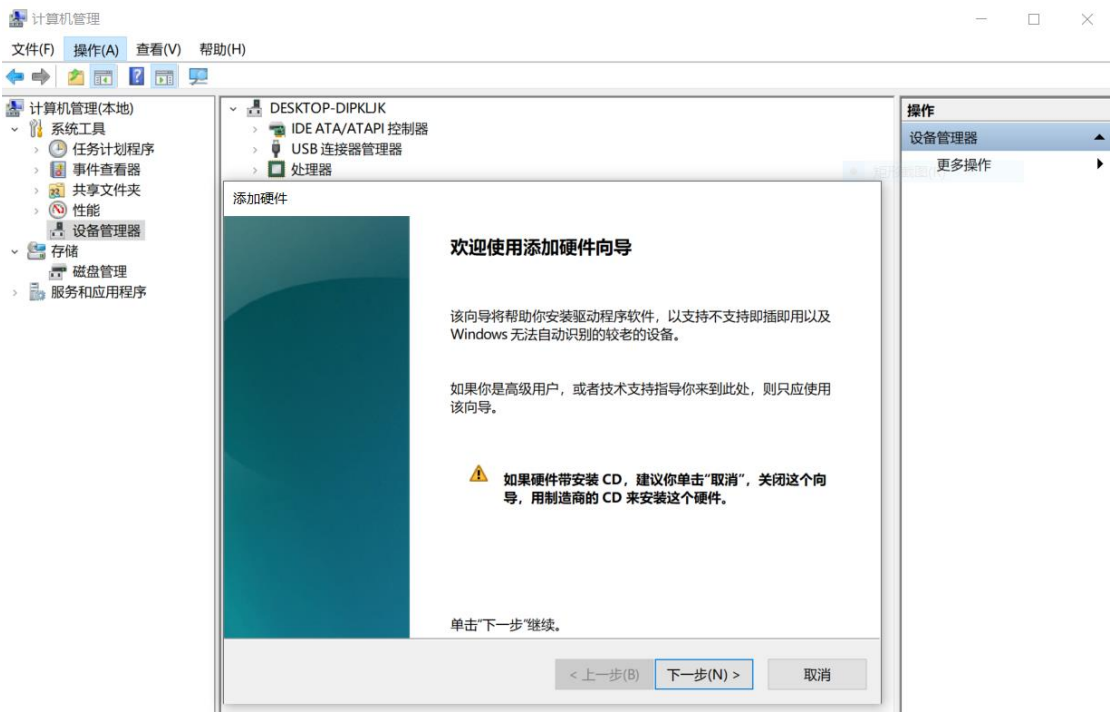
57118109 徐一鸣

实验 0 准备工作

先不启动虚拟机,为了保证主机能通过 IP 访问虚拟机,需要更改连接方式为桥接网络, virtualbox-网络-网卡 1-桥接网络, 选择 Microsoft KM-TEST 环回适配器:



在“计算机”图标上右键选择“管理”，在打开的“计算机管理”窗口中选择左侧的“设备管理器”，点击“操作”再点击“添加过时硬件”，点击“下一步”。



选择“安装我手动从列表选择的硬件”。

添加硬件

这个向导可以帮助你安装其他硬件

这个向导可以搜索其他硬件并为你自动安装。或者，如果你知道要安装哪个型号的硬件，你可以从列表选择。

你想向导做什么？

- ☒ 搜索并自动安装硬件(推荐)(S)
- ☐ 安装我手动从列表选择的硬件(高级)(M)

< 上一步(B)

下一步(N) >

取消

选择“网络适配器”。

添加硬件

从以下列表，选择要安装的硬件类型

如果看不到想要的硬件类型，请单击“显示所有设备”。

常见硬件类型(H):

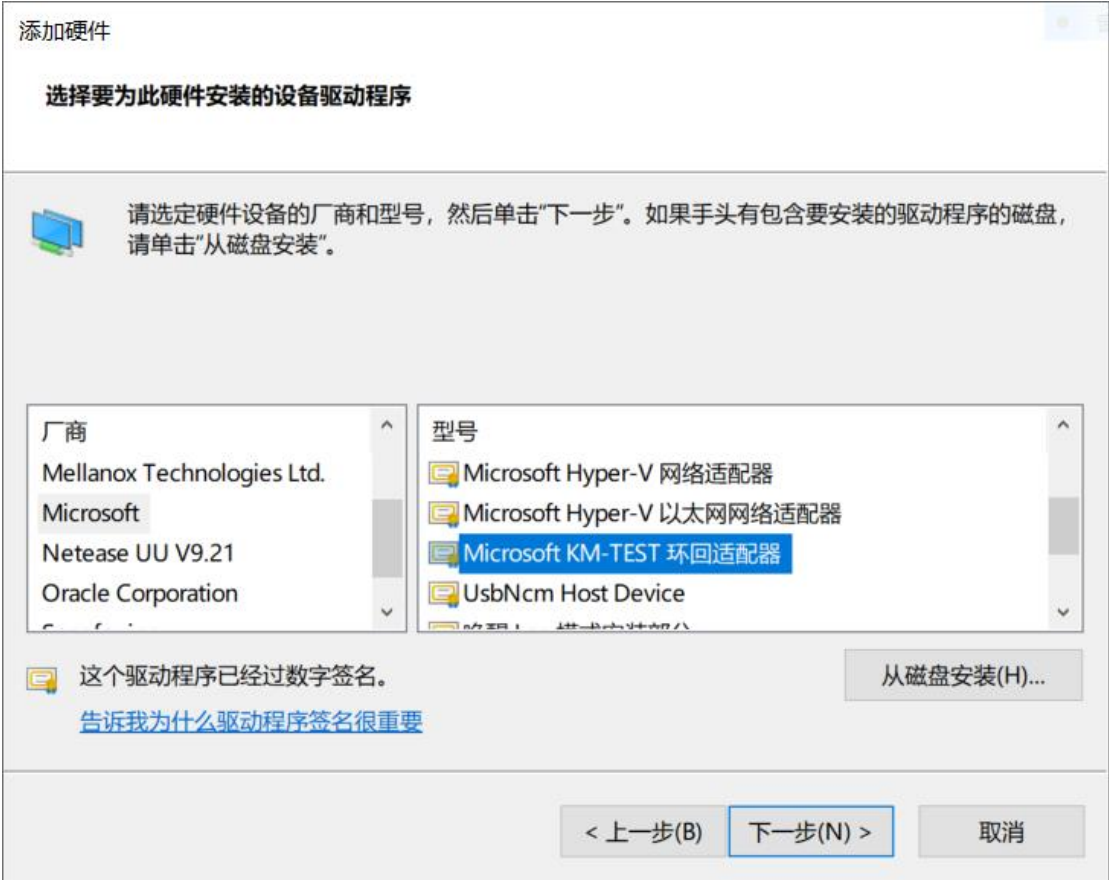
-  通用远程桌面设备
-  图像设备
-  网络适配器
-  系统设备
-  显示适配器
-  远程桌面摄像头设备
-  照相机
-  智能卡

< 上一步(B)

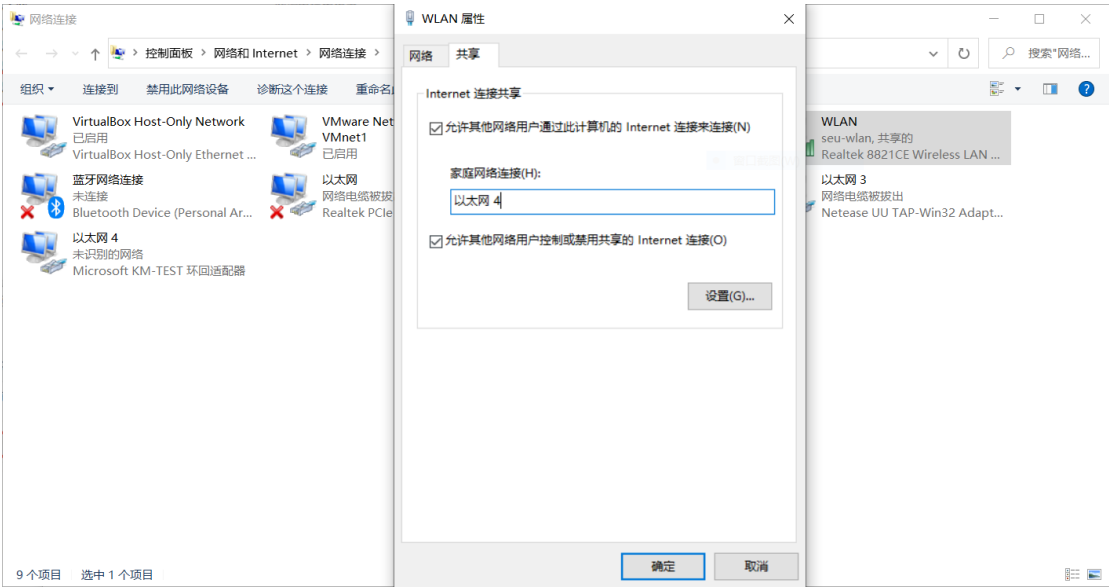
下一步(N) >

取消

厂商选择微软，型号选择 KM-TEST 环回适配器 此时在网络适配器中会多出一个以太网 4。



设置物理无线网卡的网络共享。在无线网卡的属性对话框中，选择“共享”标签，按图中所示设置即完成了无线网卡的共享。



在主机上进行 ping 命令的访问测试：

```
C:\Users\14532>ping 192.168.137.130

正在 Ping 192.168.137.130 具有 32 字节的数据:
来自 192.168.137.130 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.137.130 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.137.130 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.137.130 的回复: 字节=32 时间<1ms TTL=64

192.168.137.130 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 = 0ms, 平均 = 0ms
```

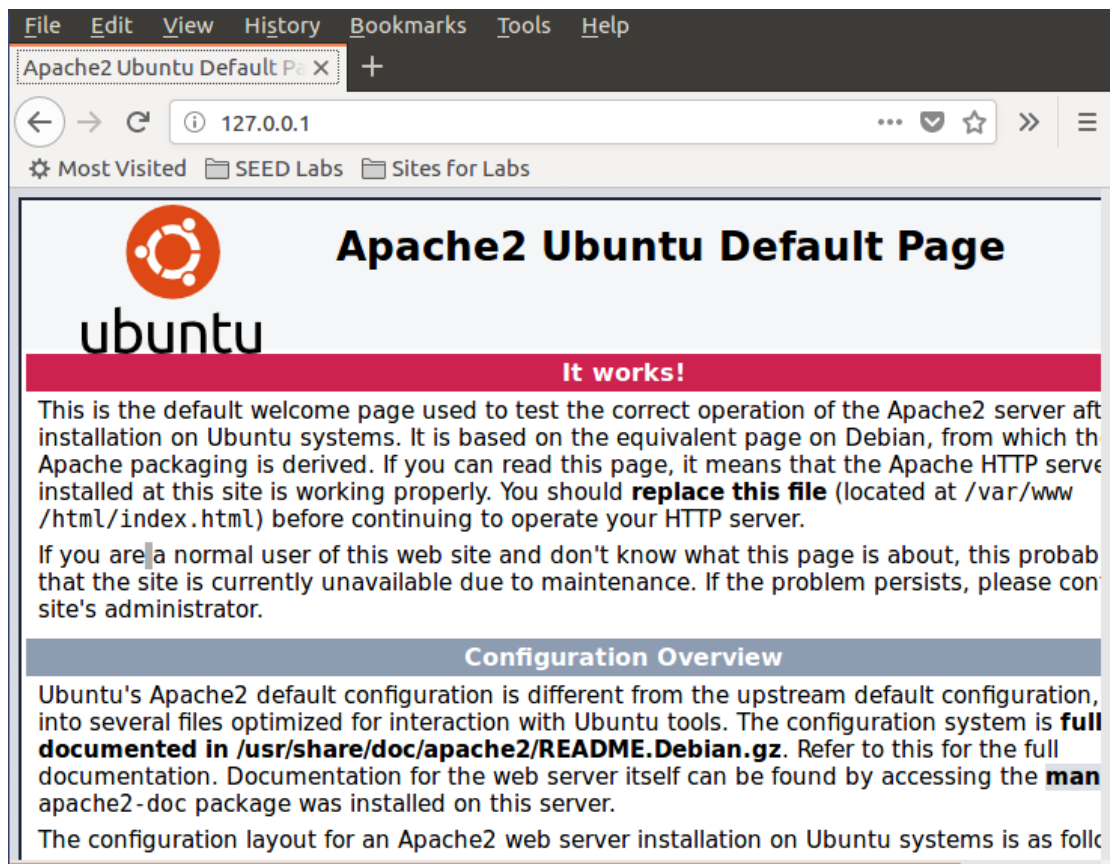
实验 1 HTTP 基础

任务一：安装 apache 服务器 并用简单页面验证

步骤 1：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 3 not upgraded.
[09/08/20]seed@VM:~$
```

步骤 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。



步骤 3：

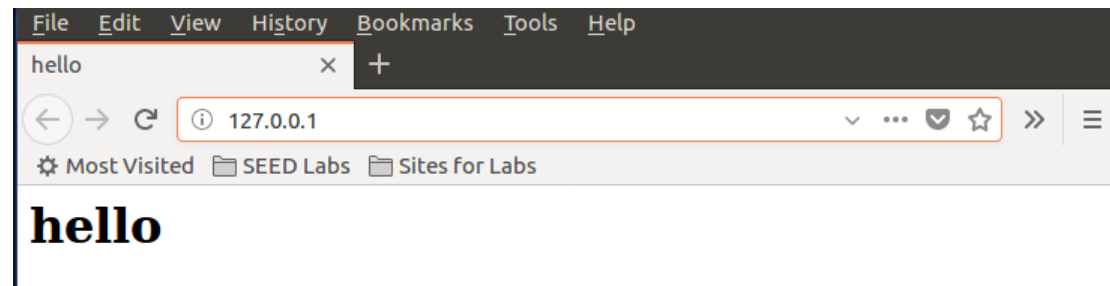
```
1. cd /var/www/html
```

2. 使用 `sudo gedit index.html` 指令打开 `index.html`，删除原内容并重新进行编写。

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

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

步骤 4：修改后使用浏览器登录 127.0.0.1，刷新后发现页面更改为新主页。



任务二：通过 host 文件解析名称

步骤 1：虚拟机内输入 `ifconfig` 查询虚拟机 ip 地址，以管理员身份打开记事本程序，记事本中打开 windows 主机中的 `hosts` 文件（`C:\Windows\System32\drivers\etc`），`hosts` 文件加入虚拟机 ip 地址与主机名 `vulnerable` 并保存。

```
[09/08/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:8a:4d:69
          inet addr:192.168.137.130  Bcast:192.168.137.255  Mask:255.255.255.0
          inet6 addr: fe80::8069:ccbc:53dd:af0d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2552 (2.5 KB)  TX bytes:9828 (9.8 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:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21989 (21.9 KB)  TX bytes:21989 (21.9 KB)

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


此电脑 > Windows (C:) > Windows > System32 > drivers > etc

名称	修改日期	类型	大小
hosts	2020/8/21 1:31	文件	1 KB
lmhosts.sam	2019/3/19 12:49	SAM 文件	4 KB
networks	2018/9/15 15:31	文件	1 KB
protocol	2018/9/15 15:31	文件	2 KB
services	2018/9/15 15:31	文件	18 KB

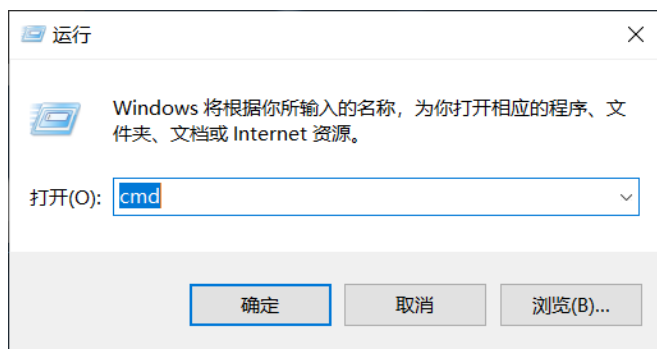
```
*hosts - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by 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
#192.168.137.130 vulnerable
192.168.137.130 vulnerable
```

第 23 行, 第 28 列 100% Windows (CRLF) UTF-8

任务三：编写 HTTP 客户端，使用 http 库检索站点的主页

步骤 1: windows 主机中输入 curl+虚拟机 ip 地址可查看编写的 index 文件内容。



```
C:\Users\14532>curl 192.168.137.130
<html>
<head>
<title>hello</title>
</head>
<body>
<h1>hello</h1>
</body>
</html>
```

步骤 2: 虚拟机中输入 `python3 --version` 查看虚拟机是否有 python3.5 (本次实验大家用的虚拟机已安装 python2.7 与 python3.5 版本, python3 版本才是我们用的, 后续我们直接使用 python3 命令执行文件)。

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

步骤 3: 创建 .py 的 python 执行文件 (文件内容不限, 这里沿用实验 2 的 exploit.py)。

```
[09/08/20]seed@VM:~$ gedit exploit.py
```

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;
}

```

步骤 2: 执行该文件, 查看网页定向是否正确。

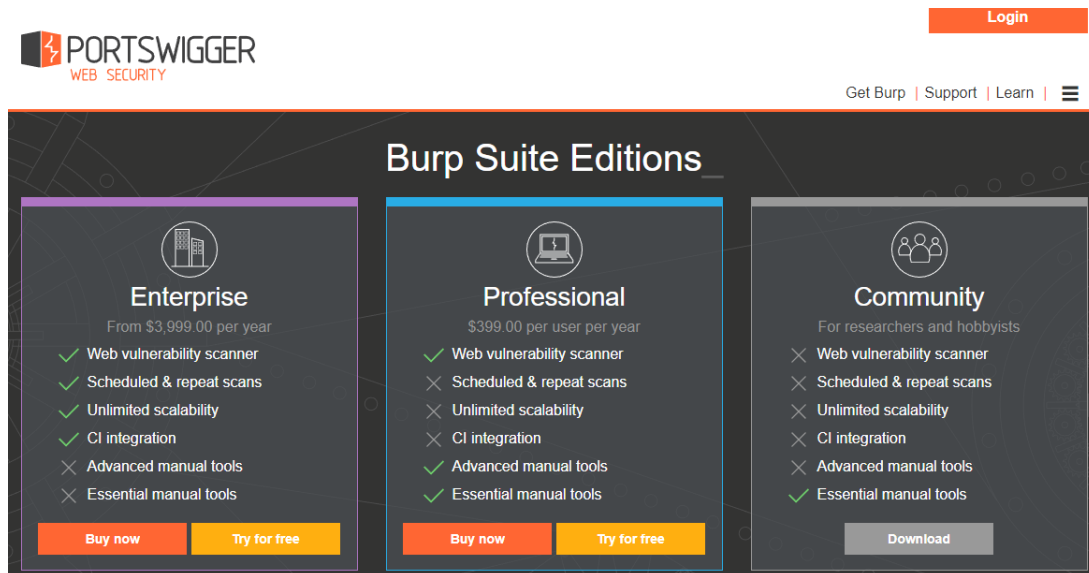

```
C:\Users\14532\source\repos\Project2\Debug\Project2.exe
HTTP/1.1 200 OK
Date: Wed, 09 Sep 2020 01:59:28 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Tue, 08 Sep 2020 16:24:47 GMT
ETag: "51-5aecfc60667a0"
Accept-Ranges: bytes
Content-Length: 81
Vary: Accept-Encoding
Connection: close
Content-Type: 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. . .
```

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 1: 从 <https://portswigger.net/burp> 网站中下载 Comuunity 版本(需配置 jdk 环境)。

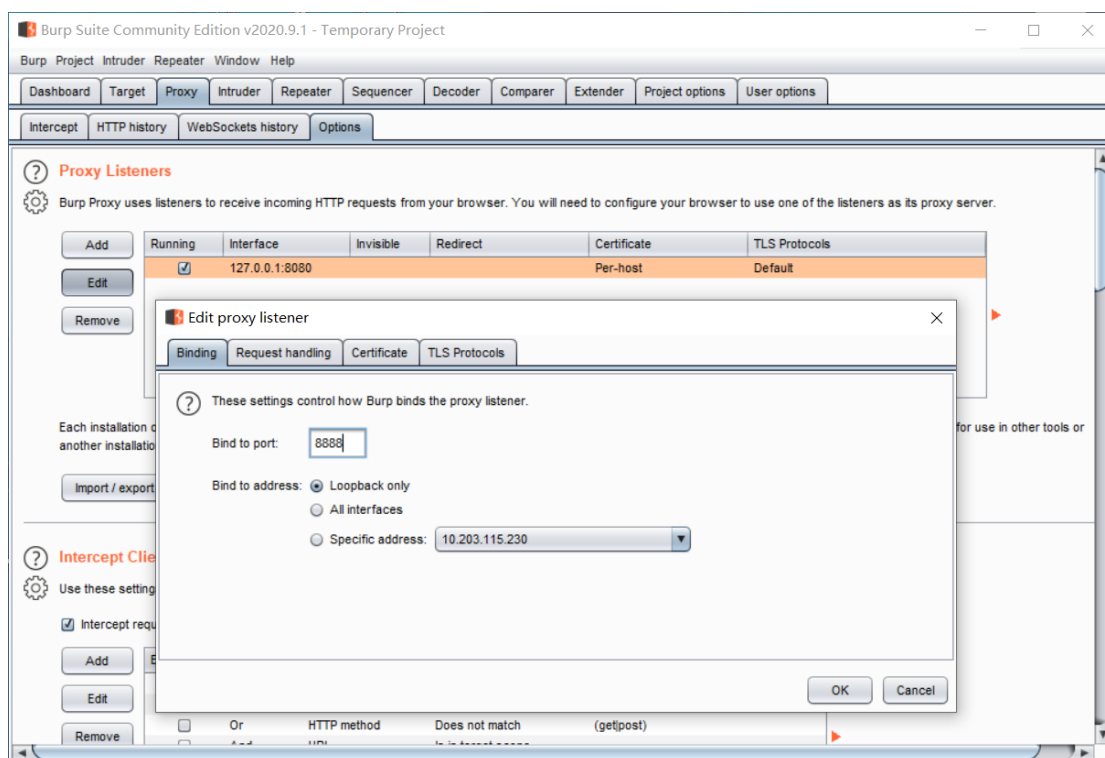


步骤 2: 对测试浏览器 Chrome 进行代理设置, 地址设为 127.0.0.1, 端口修改为 8888。





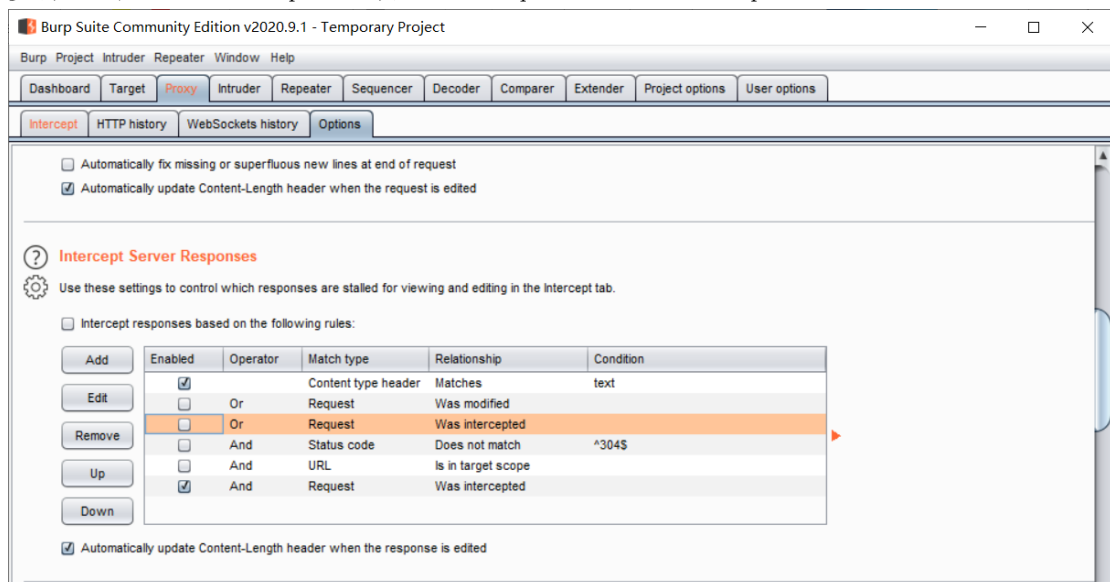
步骤 3: 打开 Burp Suite 界面，设置 Proxy 代理，端口改为 8888（需安装 CA 安全证书）。



步骤 4: 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，输入网址后会有信息，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）



步骤 5: 在 Server Response 添加 Ans-Request-Was intercepted。



步骤 6: 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)，在 csdn 登录页面点击忘记密码，出现如图。

CSDN

找回密码

+ 86

18012308828

6位数字验证码

获取验证码

新密码为11-20位数字字母组合

👁

其他找回密码方式

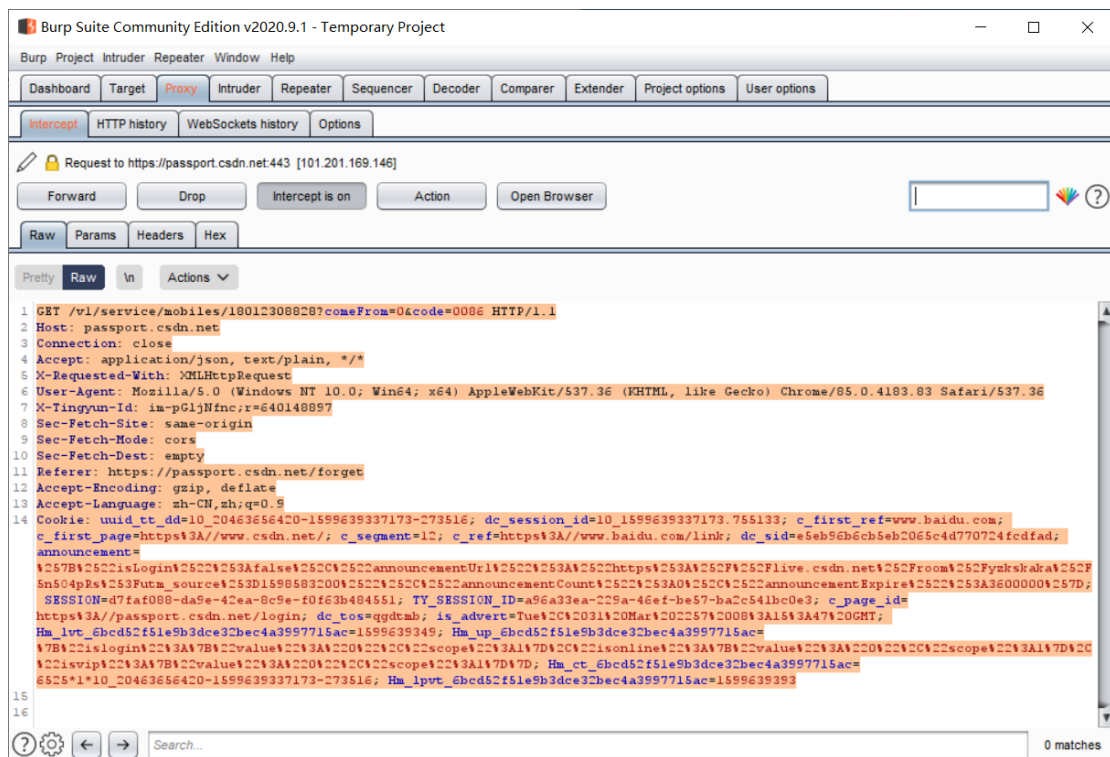
1) 用户名 找回

2) 邮箱 找回

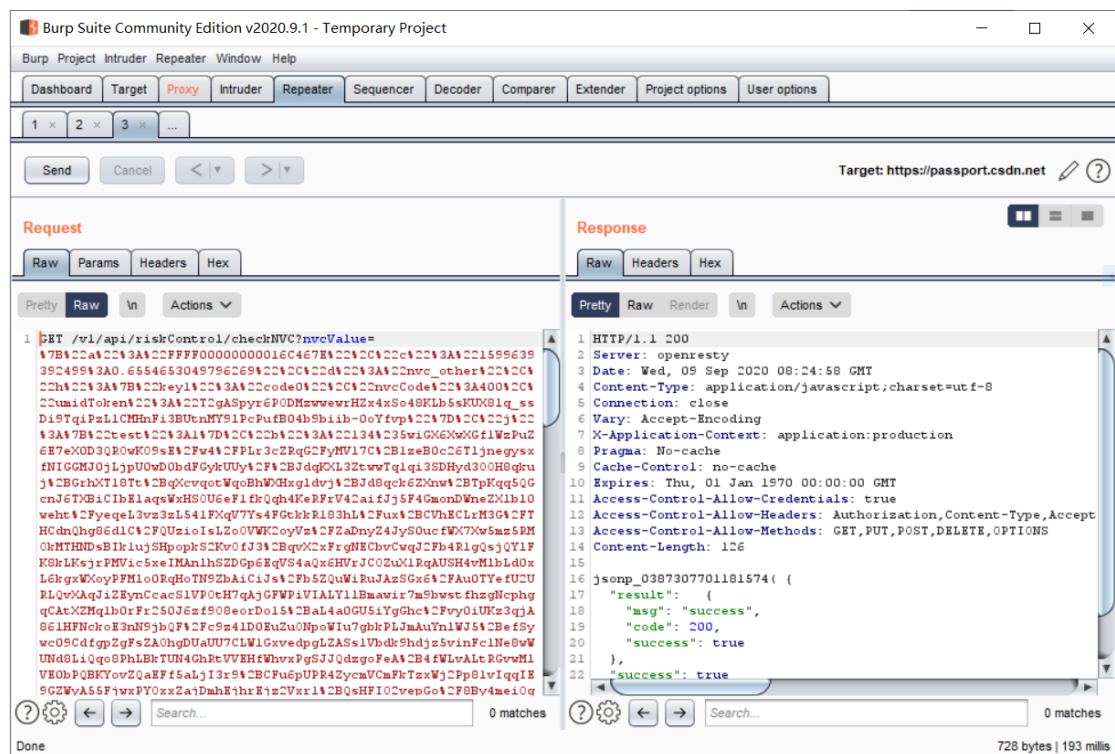
3) 手机号或邮箱不可用?

下一步

点击获取验证码，用 burp 进行拦截请求，全选并右击选择 Send to Repeater，切换到 repeater 选项面板，点击 send 发送篡改的请求，得到响应。



响应结果为 success，说明发送验证码成功，该用户电话号码有效。



实验感想

本次试验需要在桥接模式下进行，由于虚拟机先前没有进行静态 ip 地址配置，在进行 ping 命令的访问测试时遇到了瓶颈，需要在主机上进行以太网的添加，且主机与虚拟机的型号相同，才让虚拟机获得静态 ip 地址，双方才能够在桥接模式下实现联网。

在进行最后的抓包实验时需要注意的是，先按下 intercept on，再发送验证码，才能及时抓到需要的指令，教程中版本和最新版本 Chrome 的网络设置有出入。