# Lab6-report

**57117201王培丽**

**Linux Firewall Exploration Lab**

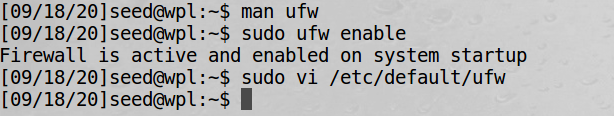
VM A 10.0.2.4

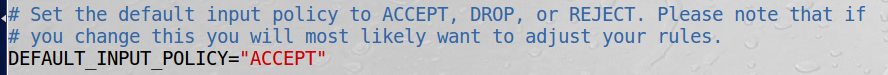
VM B 10.0.2.5

VM C 10.0.2.6

#### Task 1: Using Firewall

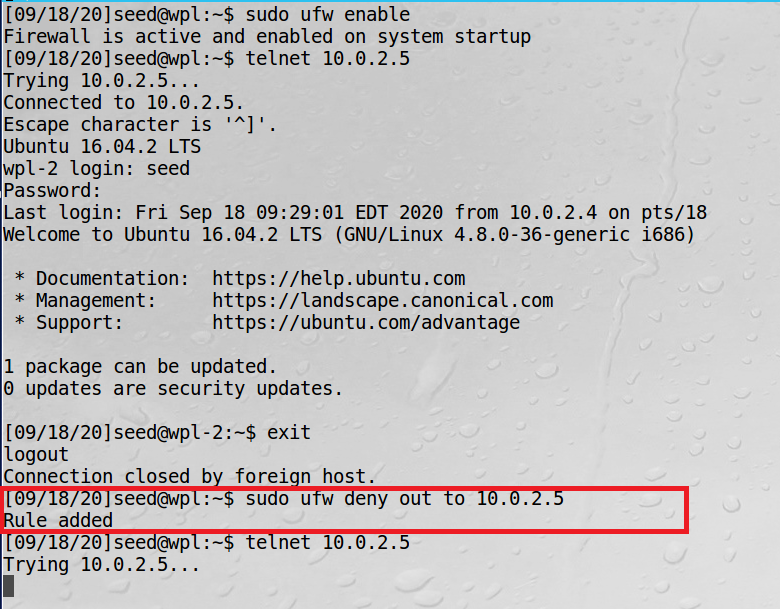
1. 在VM A上开启ufw，修改文件/etc/default/ufw：





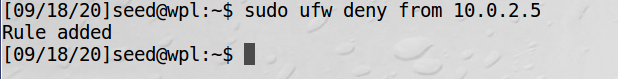
1. 阻止A(10.0.2.4) telnet连接 B(10.0.2.5)

ufw规则加入前后，telnet连接结果的不同：

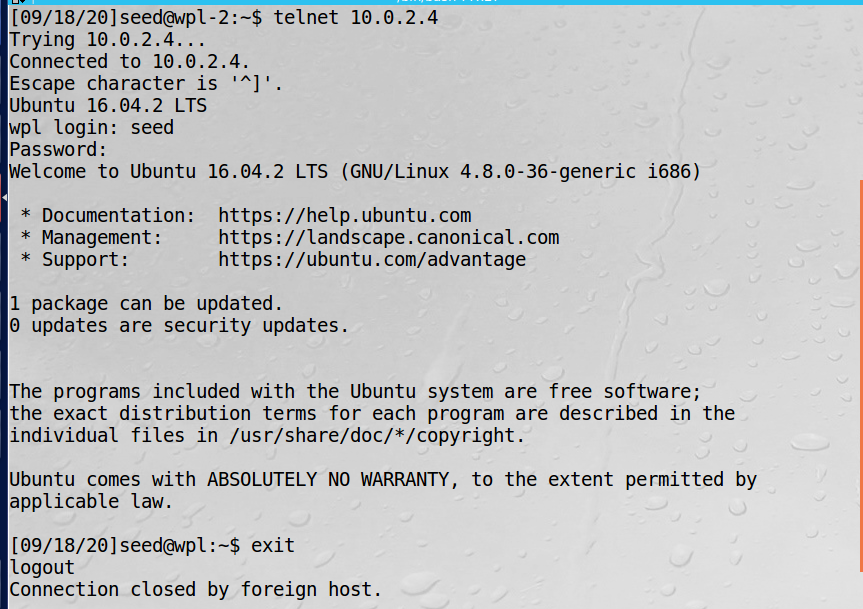


1. 阻止B(10.0.2.5) telnet连接 A(10.0.2.4)

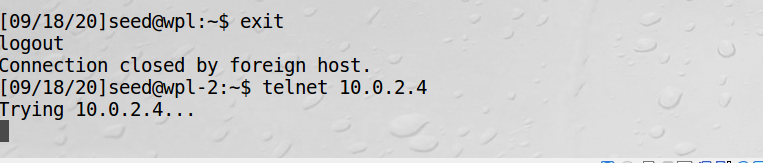
在A上添加如下规则：



添加前，B 可远程连接A：

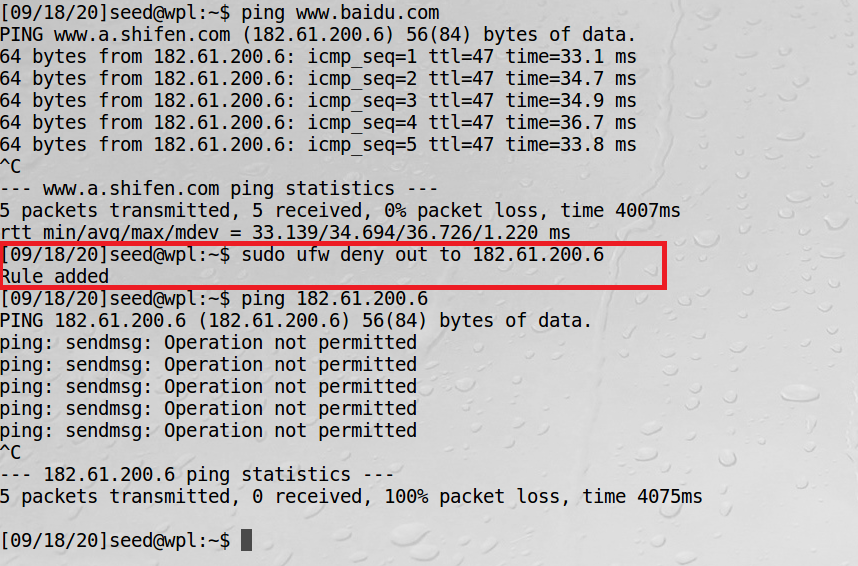


A添加规则后，B再去连接A，已经连接不上：



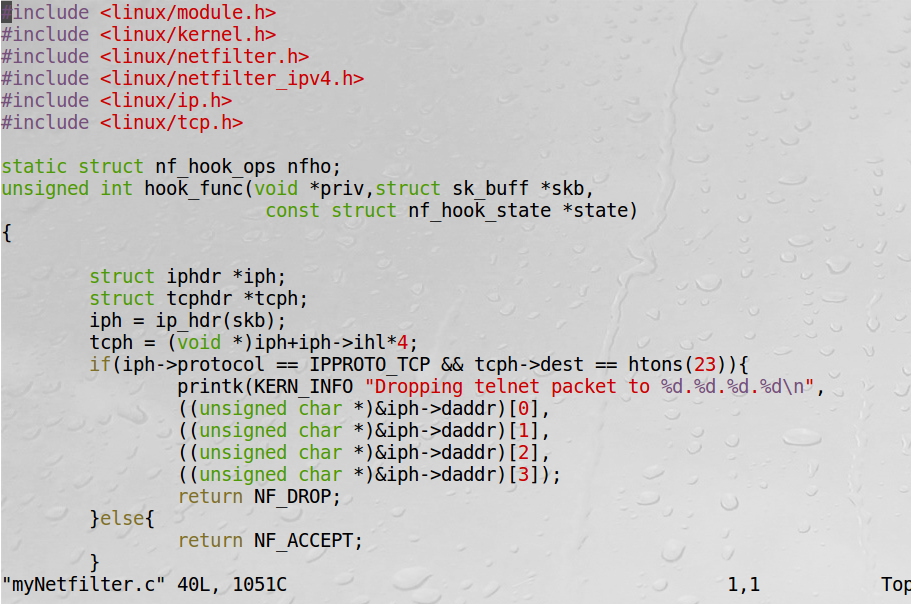
1. 阻止A访问一个外部的网站(www.baidu.com)

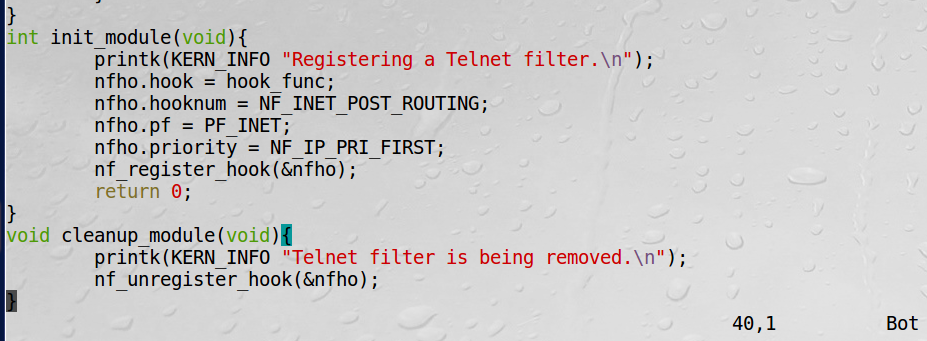
规则添加前，可ping通，添加后ping不通：



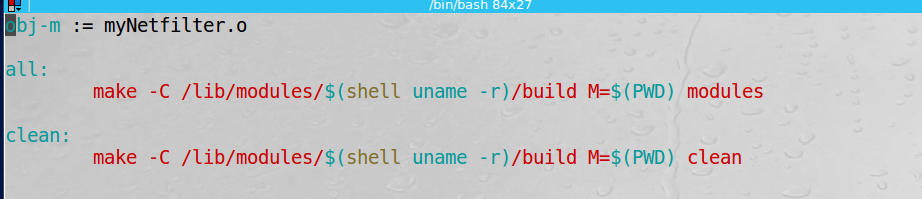
#### Task 2: Implementing a Simple Firewall

1. 编写文件myNetfilter.c，阻止telnet连接：

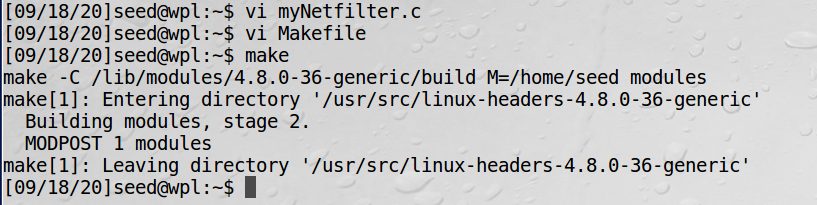




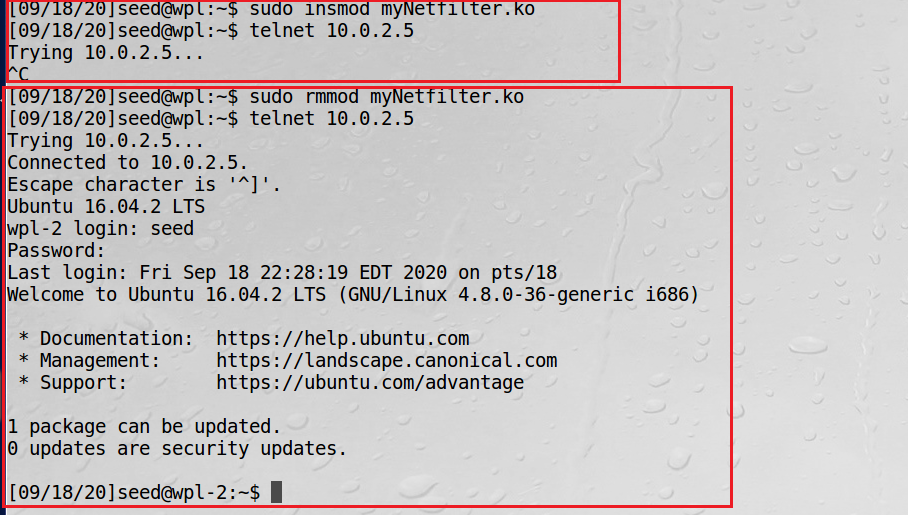
1. 编写Makefile文件：



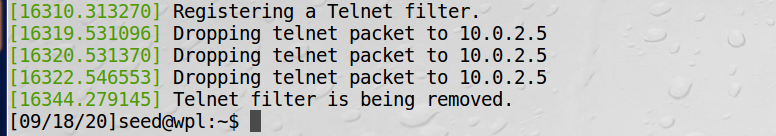
1. make



1. 添加myNetfilter.ko模块后，telnet失败，移出myNetfilter.ko模块，telnet正常：



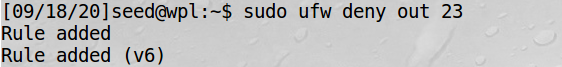
1. 查看dmesg的输出：



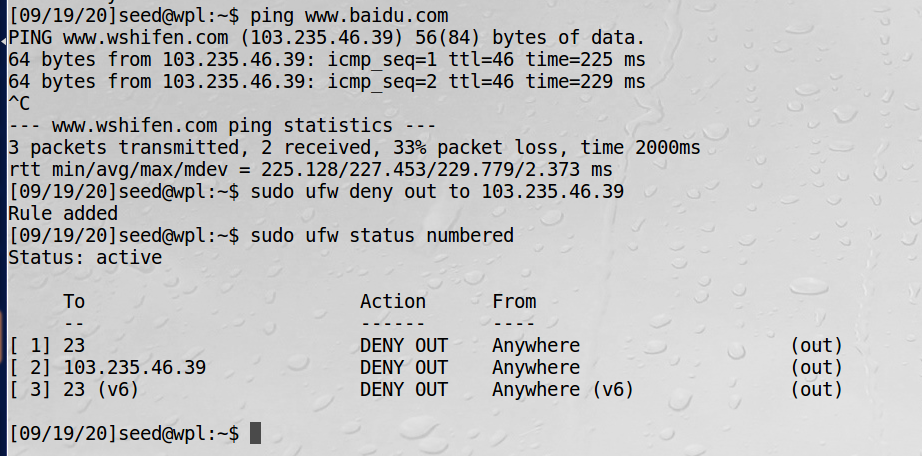
#### Task 3: Evading Egress Filtering

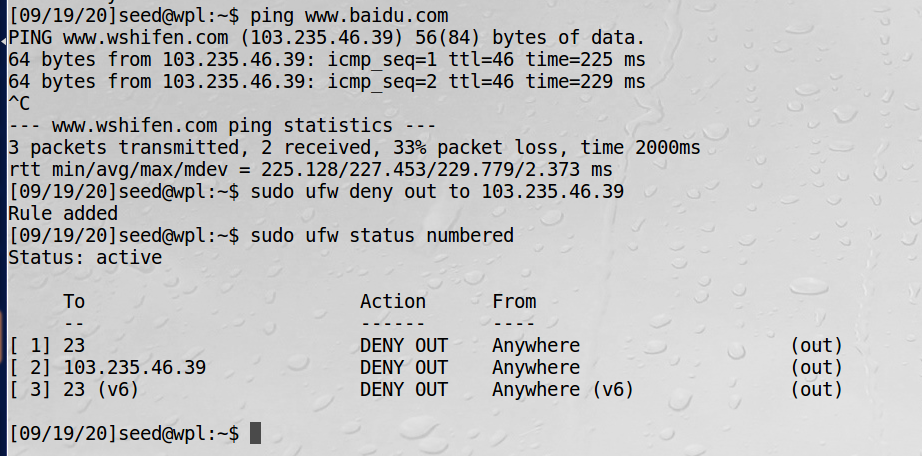
**在VM A上设置如下两条规则：**

阻止所有传出流量到外部telnet服务器：



阻止所有传出的流量到[www.baidu.com](http://www.baidu.com)：

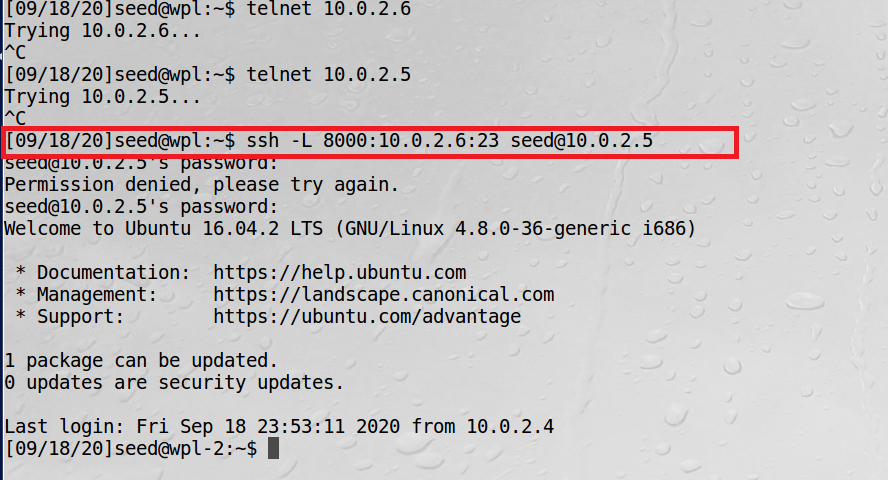




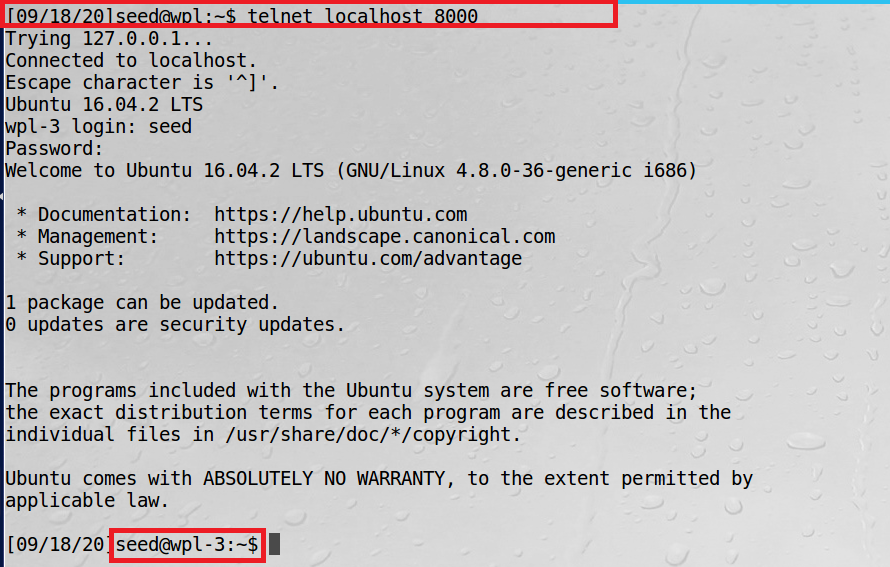
**Task 3.a: Telnet to Machine B through the firewall**

在VM A(10.0.2.4)和VM B(10.0.2.5)之间开启SSH隧道，使A的数据通过隧道到B，最终到VM C(10.0.2.6)的23号端口，即实现通过隧道远程登陆到C。

1. 开启SSH隧道前，A均不能远程连接B、C
2. 使用命令ssh -L 8000:10.0.2.6:23 seed@10.0.2.5即在本地A的8000端口与B建立隧道，到达C的23号端口



1. 之后在A的另一个终端，用命令telnet localhost 8000即可telnet连接到C：

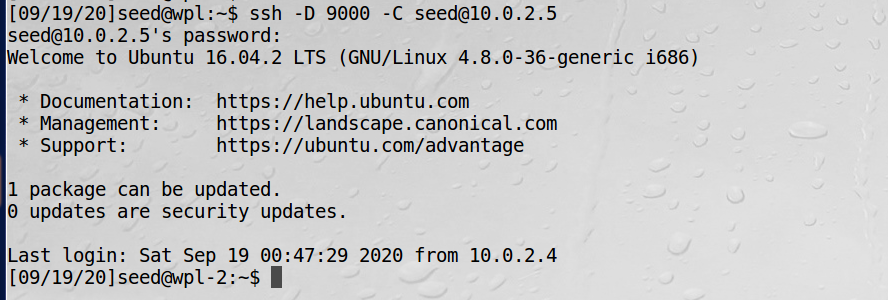


1. Wireshark中可看到10.0.2.4通过10.0.2.5访问到了10.0.2.6：

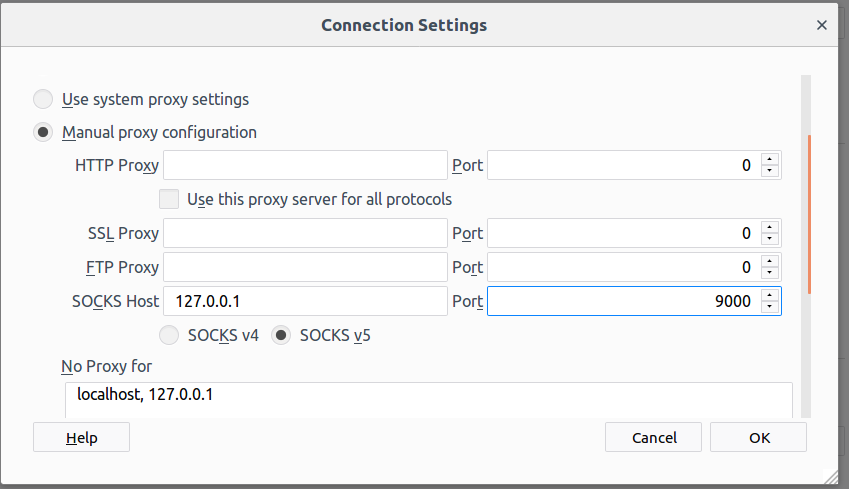


**Task 3.b: Connect to Facebook using SSH Tunnel.**

1. 使用动态端口转发方式，在A(10.0.2.4)和B(10.0.2.5)之间建立隧道：



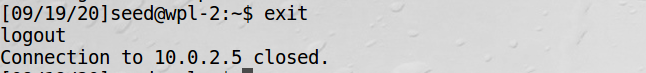
1. Firefox中配置SOCKS代理：

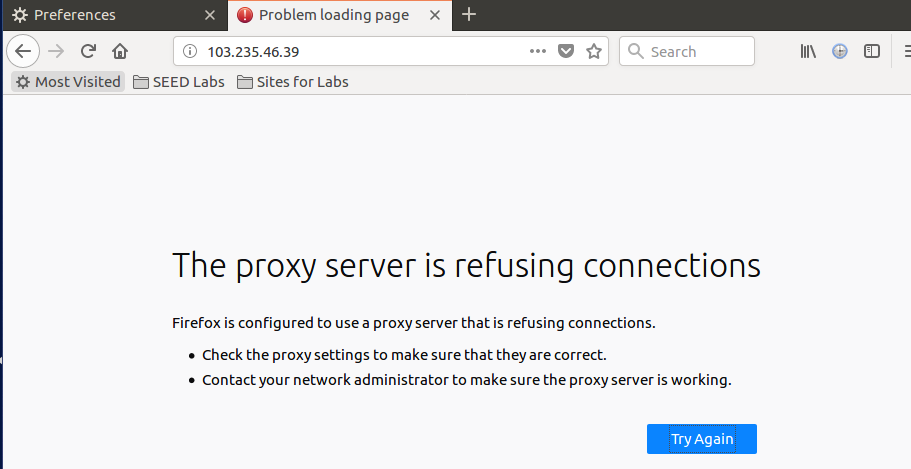


1. 启动Firefox，访问百度(103.235.46.39)，可访问：

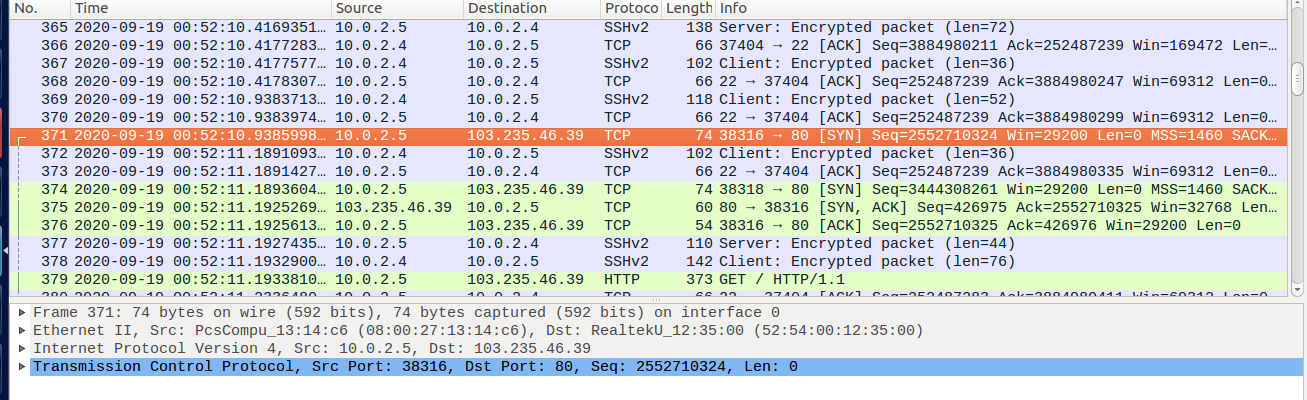


1. 断开SSH隧道，清除Firefox缓存，然后重新访问百度，已经不能访问：



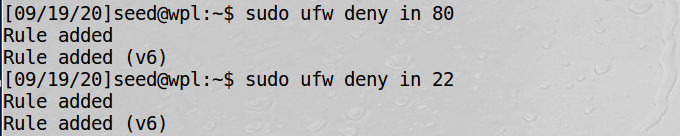


1. 再次建立SSH隧道，则又可以成功访问百度，因为主机A虽然受限，但主机B不受限，A通过B即可访问到受限的资源。
2. Wireshark中可看到10.0.2.4通过10.0.2.5实现了对103.235.46.39的访问：



#### Task 4: Evading Ingress Filtering

1. A上设置ufw规则，禁止外部访问其web服务(80端口)和ssh服务(22端口)：



1. 在A(10.0.2.4)建立到B(10.0.2.5)的反向SSH隧道：

-fCNR反向代理，使用B的9900端口和本地的22端口，登陆到B10.0.2.5的seed用户



1. 隧道建立前，B(10.0.2.5)不能通过ssh连接到A(10.0.2.4)，在B上建立对应的正向隧道：

ssh -fCNL \*:9901:localhost:9900 localhost

-fCNL正向代理，使用本地主机B的9901端口，将数据转发到9900端口，进而最终访问到A的22端口，\*表示可接受任何IP的访问

1. 即可使用命令ssh -p 9901 seed@10.0.2.5连接到A：

(登录B的9901端口->B的9900端口->A的22端口)

