

网络攻防技术基础课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 实验五  防火墙 |
| 实验日期： | 2024-04-23 |
| 实验地点： | 西部片区4号楼301 |
|  | |
| 学号： | 33920212204567 |
| 姓名： | 任宇 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2023-2024学年第二学期 |

1. **实验目的**

* 熟悉并实践常见防火墙，例如ufw(Uncomplicated FirWall)和iptables。

1. **实验用到的软件或工具**

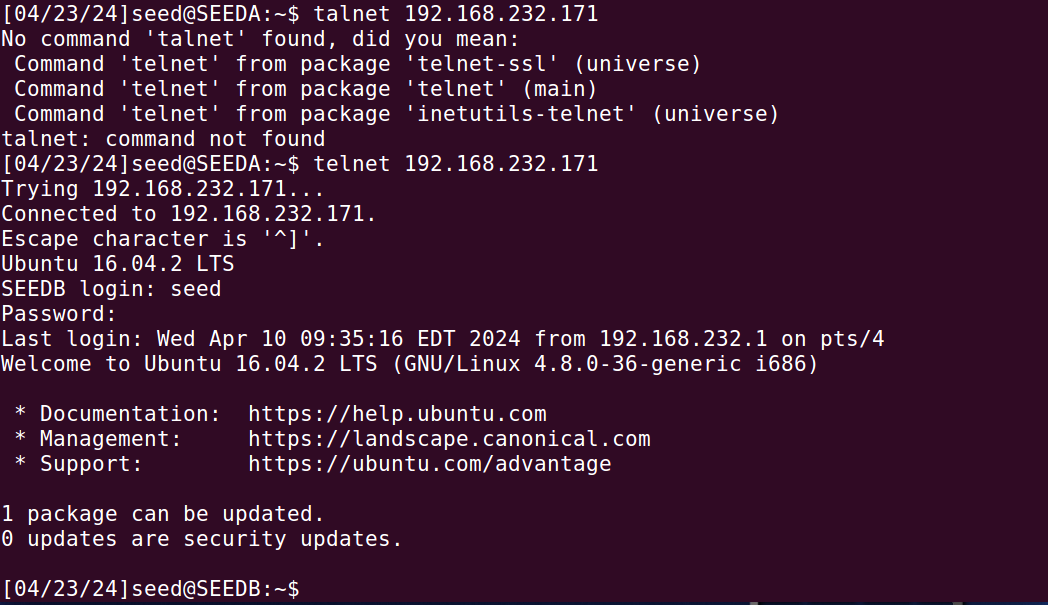
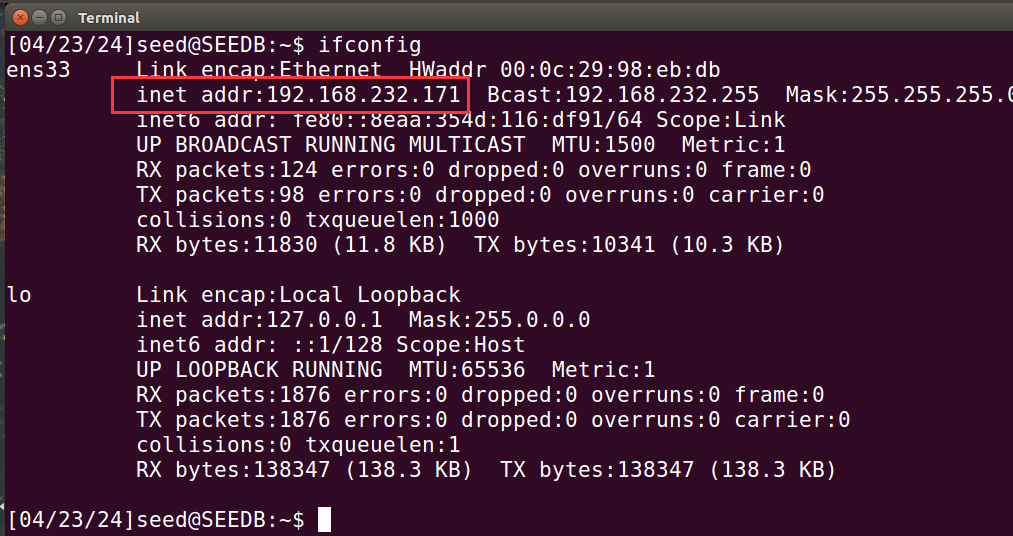
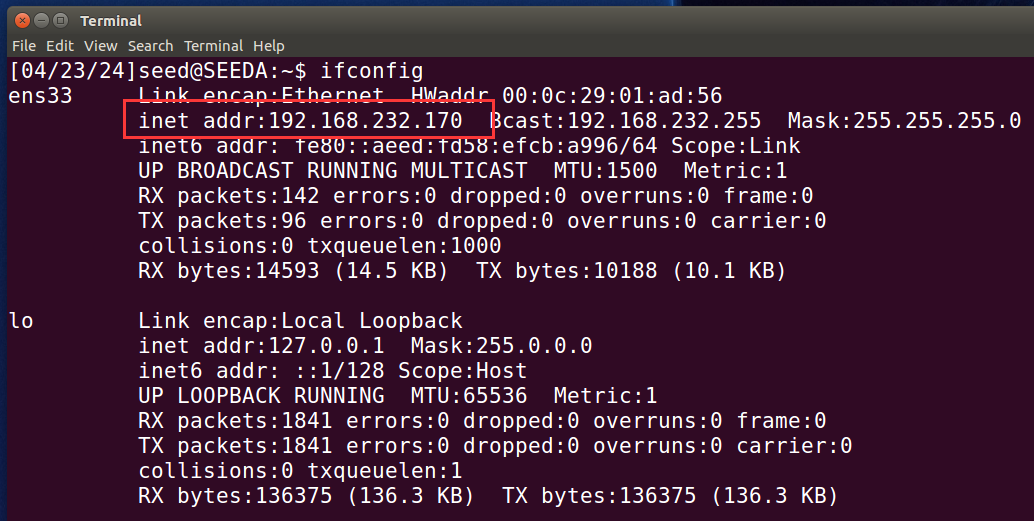
虚拟机软件：VMware Workstation Pro 17

虚拟机系统：SEED Ubuntu-16.04-32bit

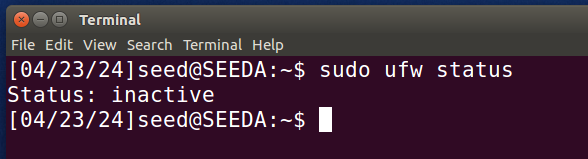
网络环境：三个虚拟机环境配置在同一NAT网络中

* SEEDA：192.168.232.170
* SEEDB：192.168.232.171
* SEEDC：192.168.232.169

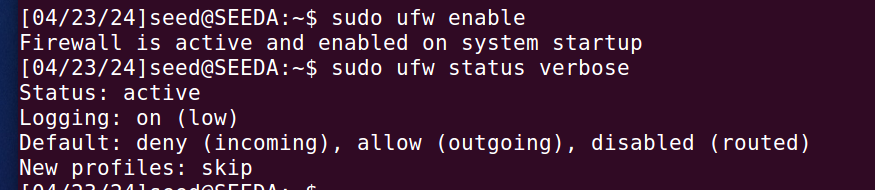
1. **实验过程**
2. **UFW使用：**
3. 首先开起两个虚拟机，用SEEDA来telnet远程连接SEEDB：



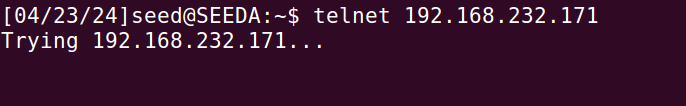
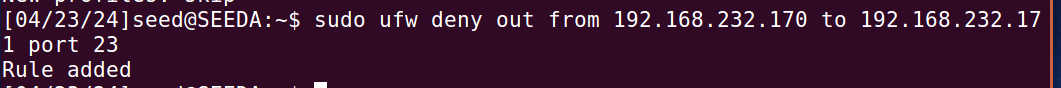
1. 可见此时的telnet可以正常连接，接着退出telnet，查看SEEDA的防火墙状态，在SEEDA下运行sudo ufw status：



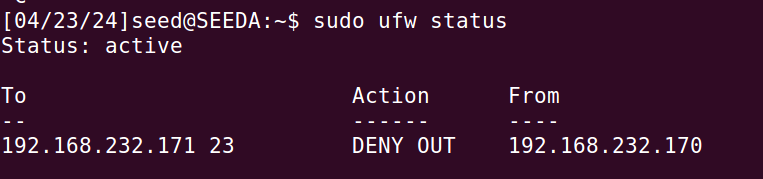
1. 配置防火墙阻止SEEDA远程连接SEEDB，在SEEDA下执行命令sudo ufw enable，并查看SEEDA的防火墙状态：



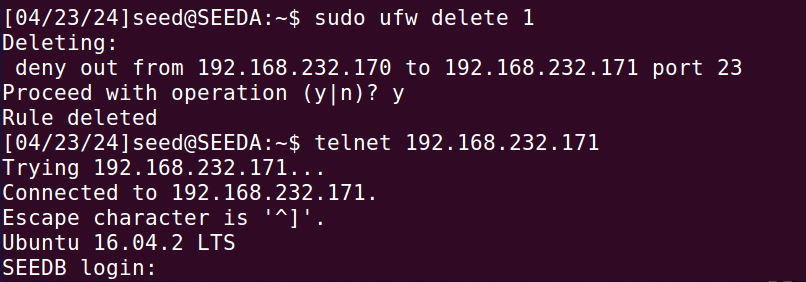
1. 增加防火墙规则，使得阻止SEEDA远程连接SEEDB，但SEEDB仍可以telnet SEEDA：

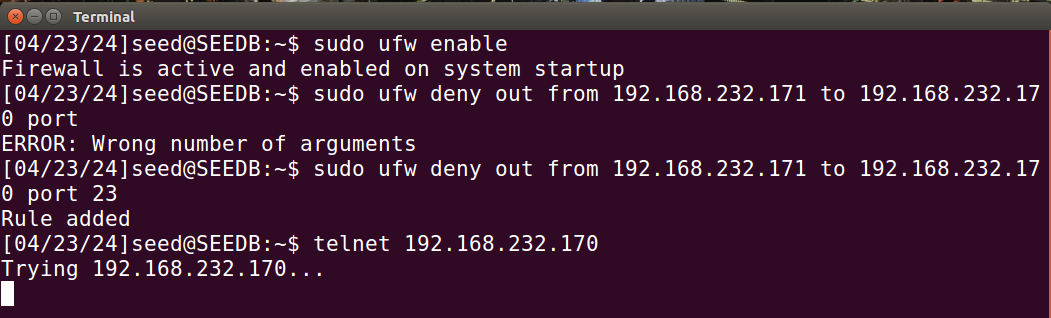


1. 查看SEEDA的防火墙状态：

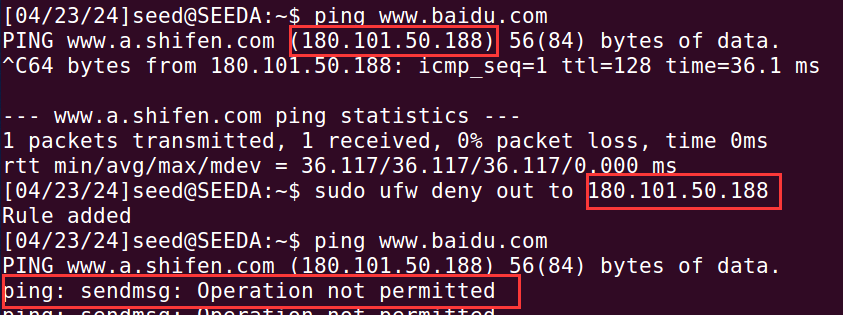


1. 删除防火墙规则，再尝试telnet，此时成功连接，同理，可以在SEEDB上配置防火墙规则阻止SEEDB远程连接SEEDA：

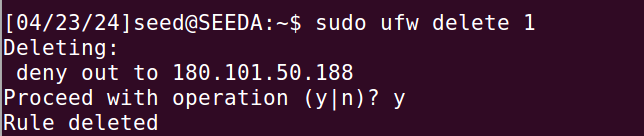




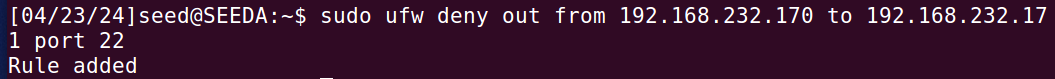
1. 配置SEEDA规则，阻止访问百度网站：

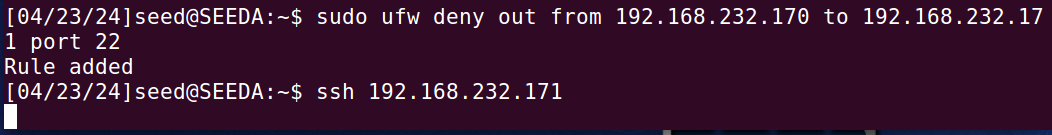


1. 删除规则后可以正常访问：

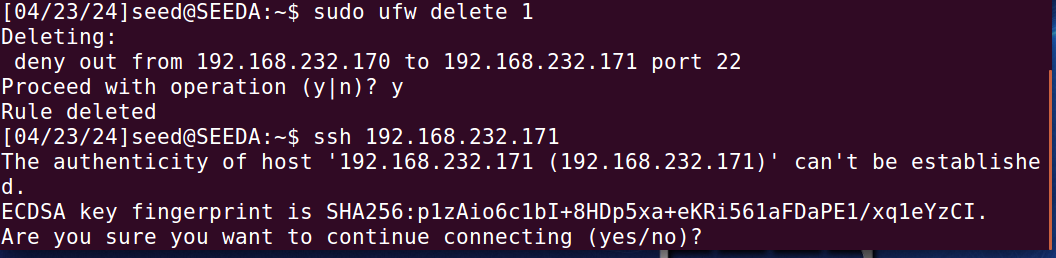


1. 同理可以阻止SEEDA的SSH连接，设置防火墙后无法建立连接：



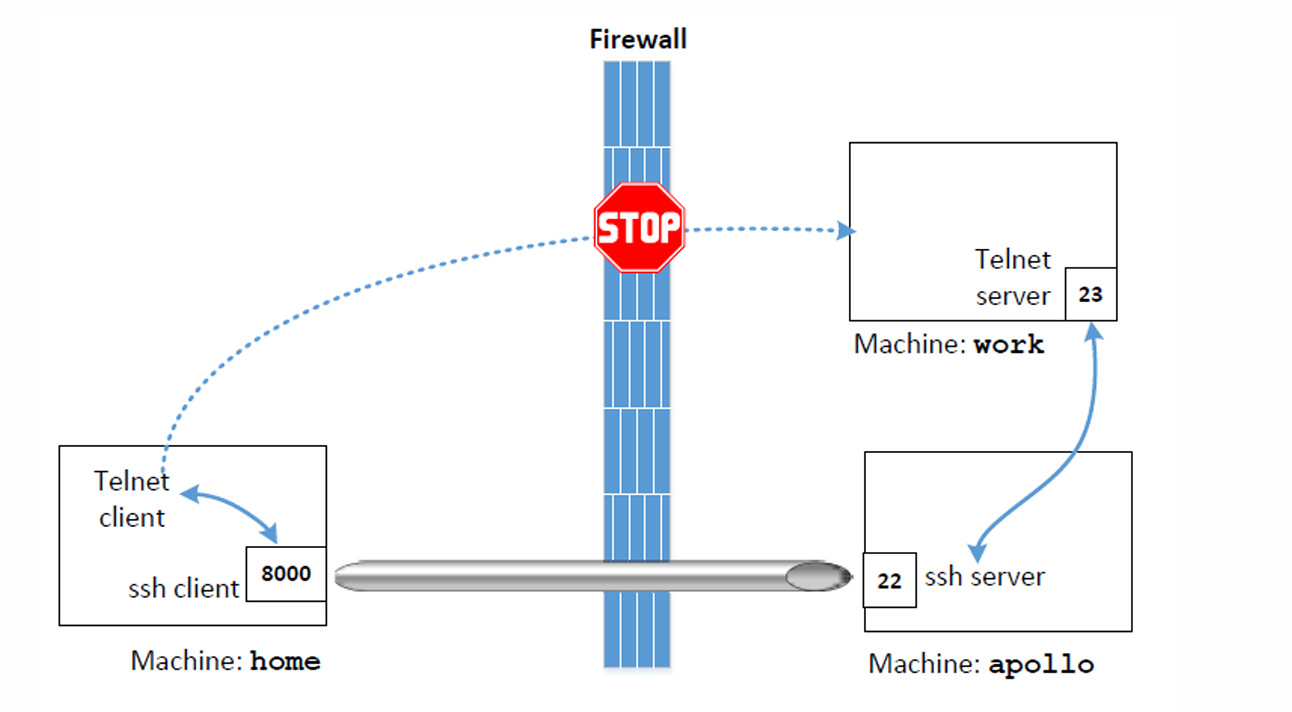


1. 删除防火墙后成功建立连接：

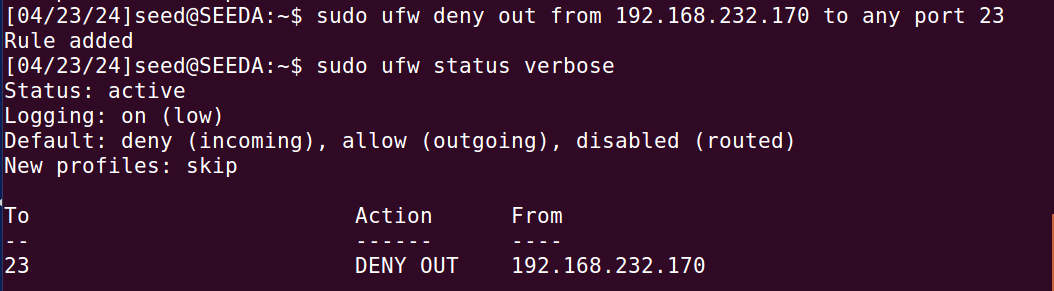


1. **绕过防火墙1：**

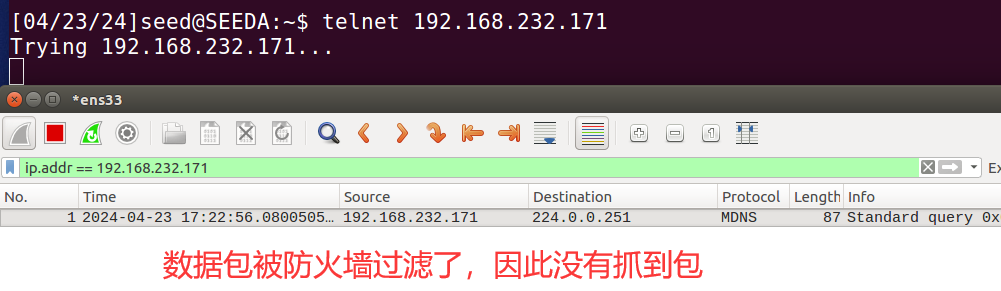
虽然防火墙阻止了相应规则的连接，但可以通过SSH隧道来绕过防火墙，示意图如下：



1. 首先阻止SEEDA远程连接其他任何主机：

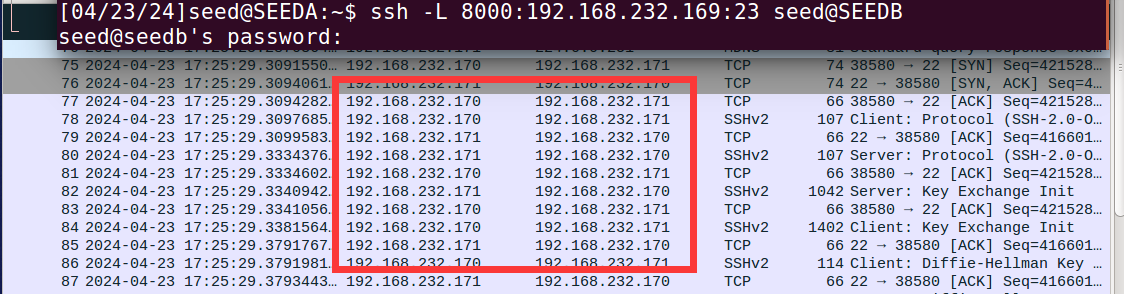


此时使用SEEDA telnet SEEDB，观察wireshark抓包，可见：

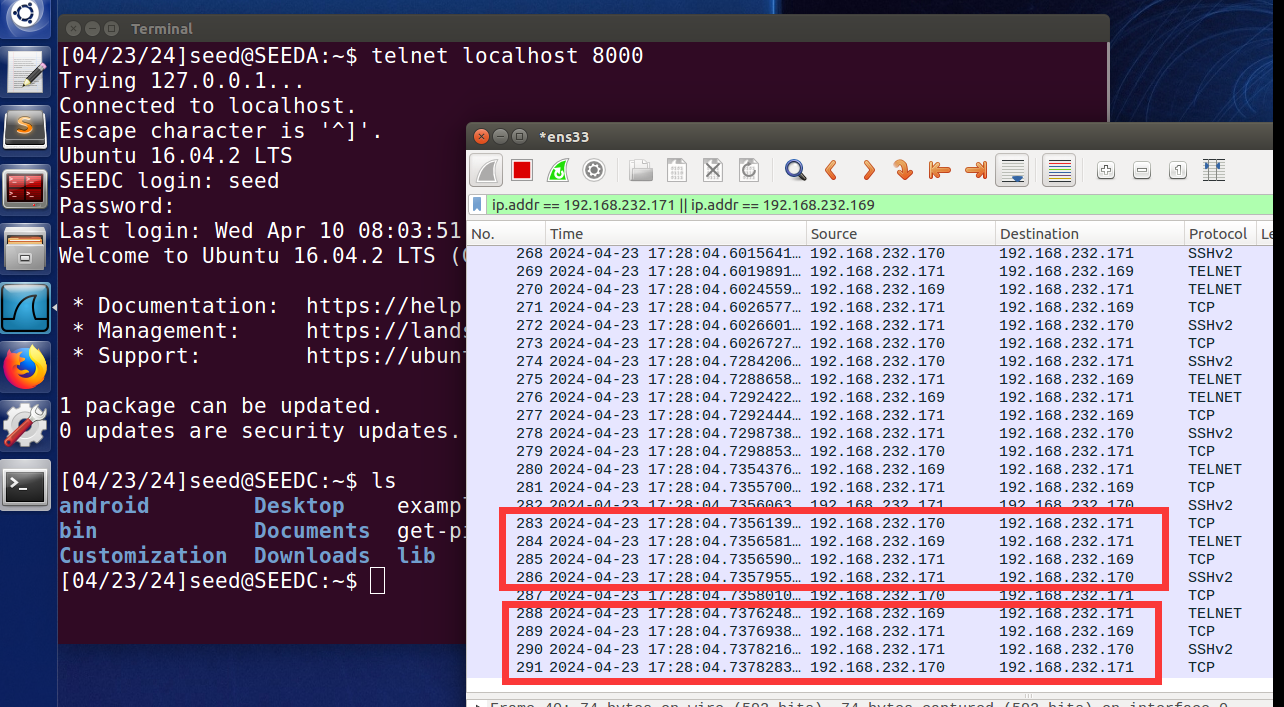
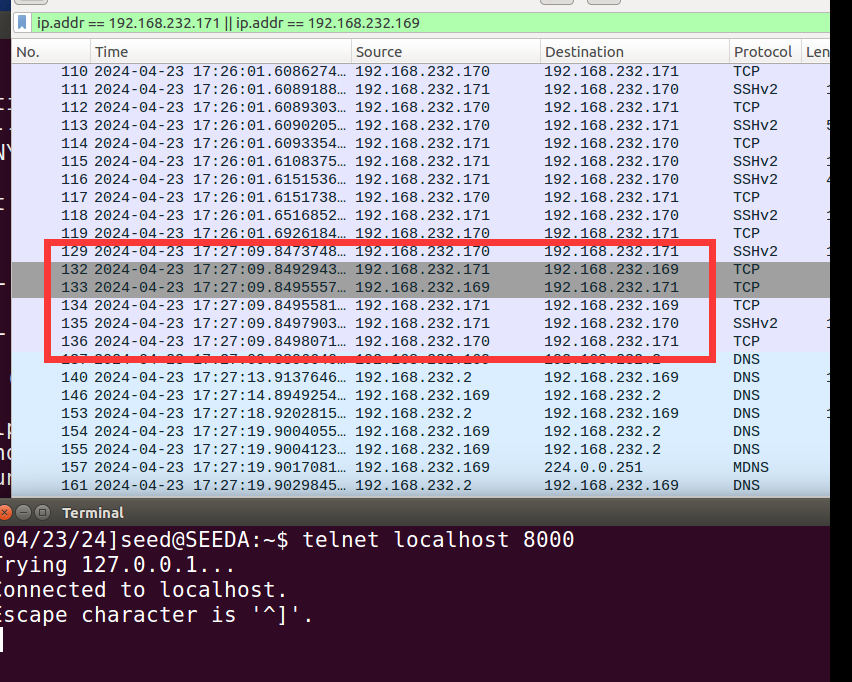


1. SEEDA和SEEDB之间建立ssh隧道，SEEDA下输入命令

ssh–L 8000: SEEDC\_ip:23 seed@SEEDB：



1. telnet localhost 8000查看是否登录到了SEEDC：



通过抓包，我们可以分析出TCP数据包的流动：

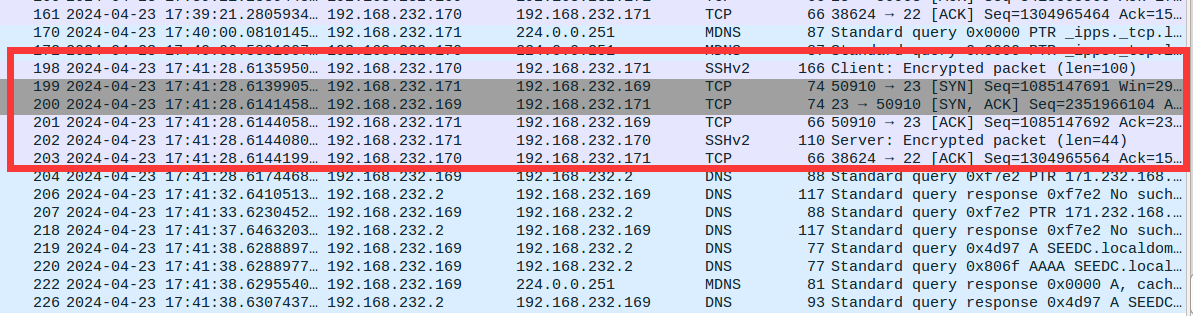
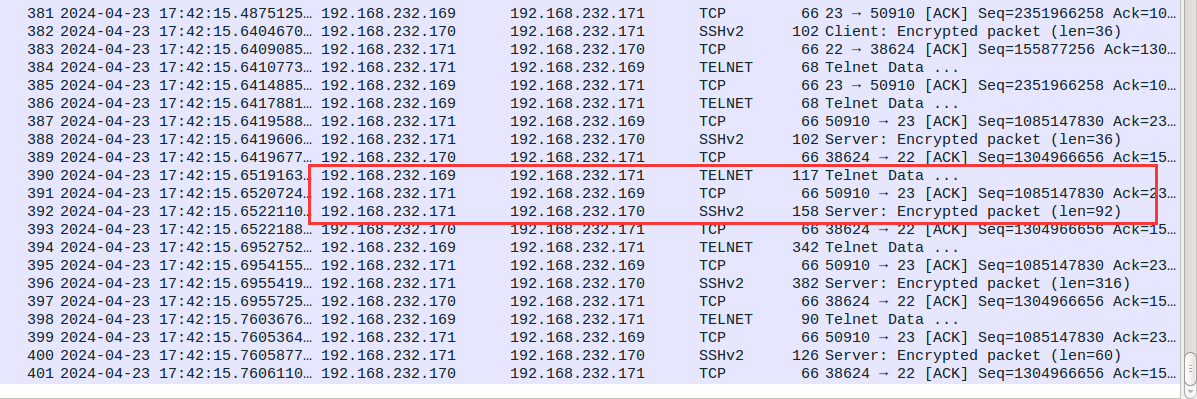
* + 建立SSH隧道（SEEDA到SEEDB）:

首先，SEEDA通过SSH建立到SEEDB的连接。在Wireshark中，通过抓包发现SEEDA向SEEDB发送TCP SYN包（连接请求），随后SEEDB应答TCP SYN-ACK包（连接确认），最后SEEDA发送TCP ACK包（完成连接建立）。这一系列包完成了TCP的三次握手过程，建立了SSH连接。接下来，通过该连接，SSH协议的数据开始传输，所有传输数据都是加密的。

* + 通过本地端口访问Telnet，从SEEDA访问 localhost:8000:

在SEEDA上执行telnet localhost 8000会创建一个到本机8000端口的TCP连接。可以发现，Wireshark显示SEEDA上的Telnet客户端发送TCP SYN到localhos，然后接收到TCP SYN-ACK，再发送TCP ACK完成连接。

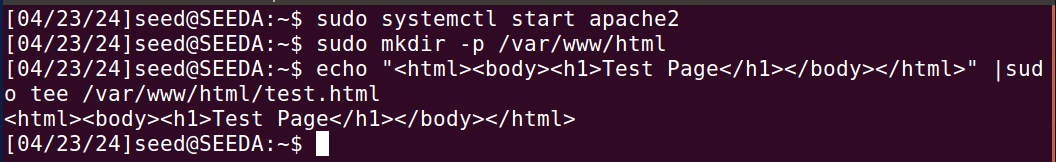
一旦连接建立，通过此连接发送的Telnet命令数据包将被封装在TCP数据段中，发送到本地的8000端口，然后通过SSH隧道转发到SEEDC的23端口。在SEEDC上，如果Telnet服务运行中，将接收到这些转发的Telnet请求，并响应相应的输出。响应同样会被封装在SSH数据包中，通过隧道回传到SEEDA。

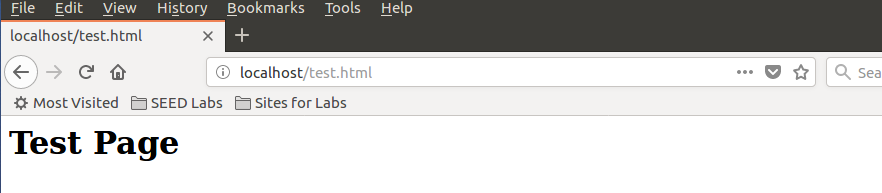
 

1. **绕过防火墙2：**

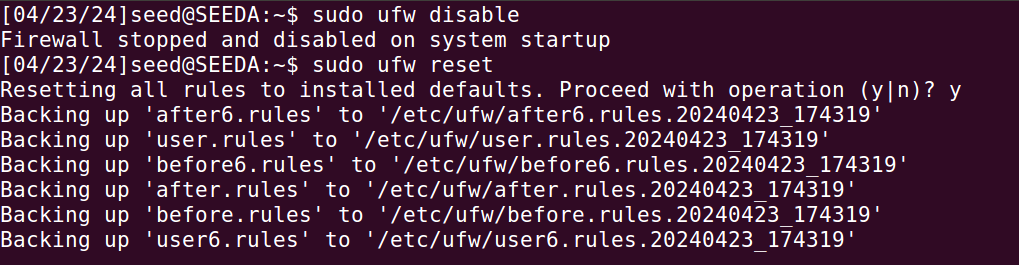
目的：SEEDA上的防火墙阻止了所有通过80端口和22端口的访问，但是SEEDA上又有Web网页（test.html）,我们需要访问该页面。

1. 在SEEDA的 /var/www/html/下建立test.html页面，并测试能够本地浏览器显示该页面：

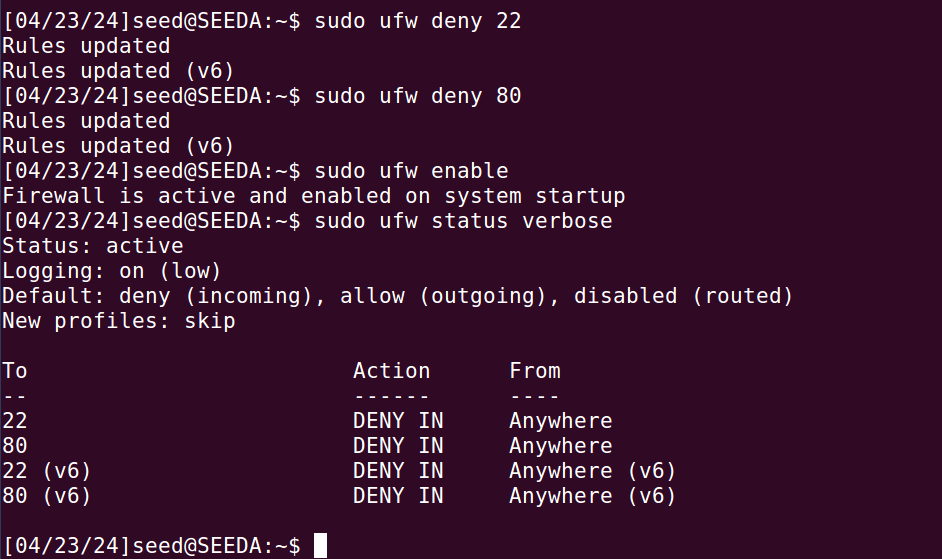




1. 删除SEEDA的所有防火墙规则：

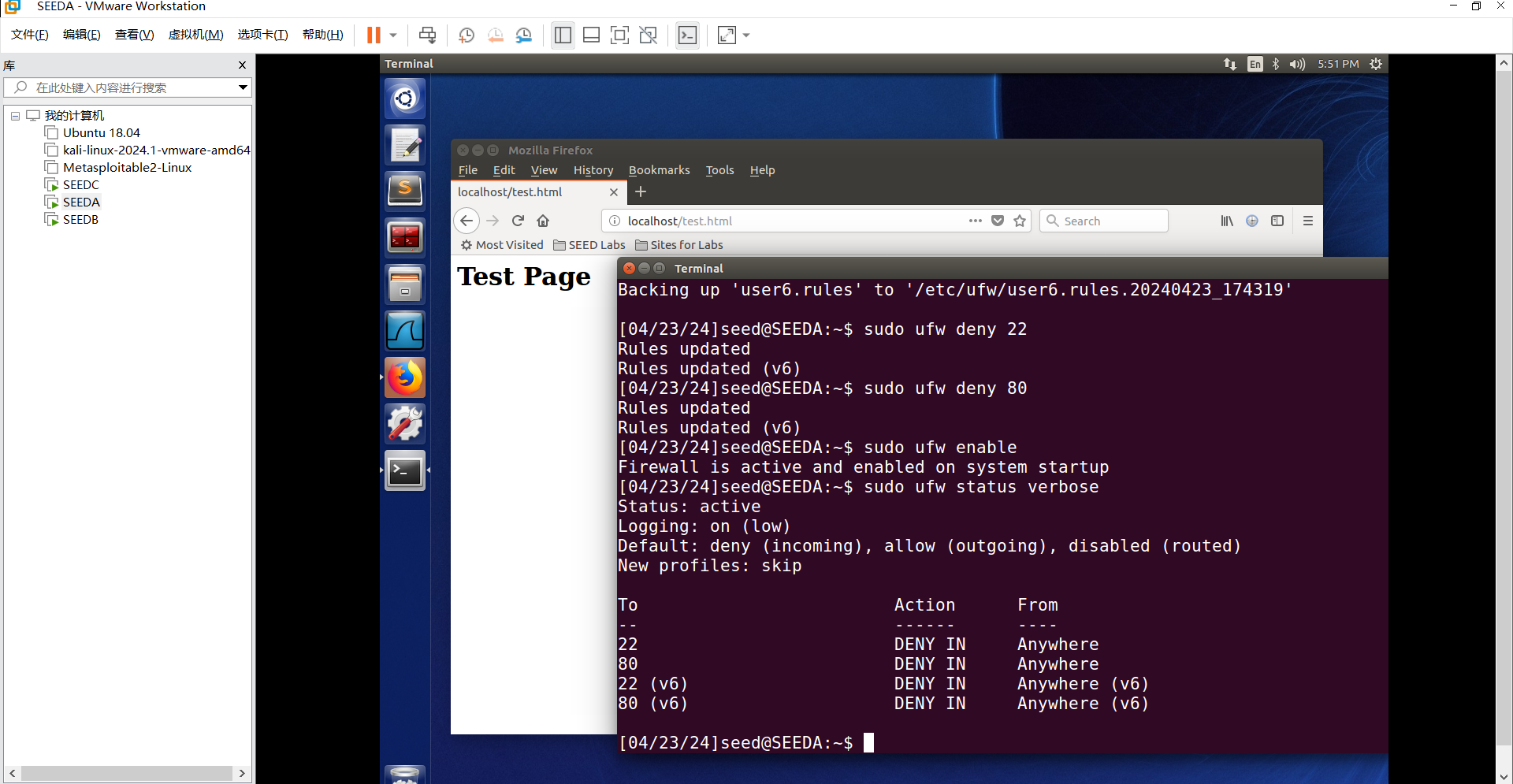


1. SEEDA下建立规则，阻止所有80和22端口的访问：

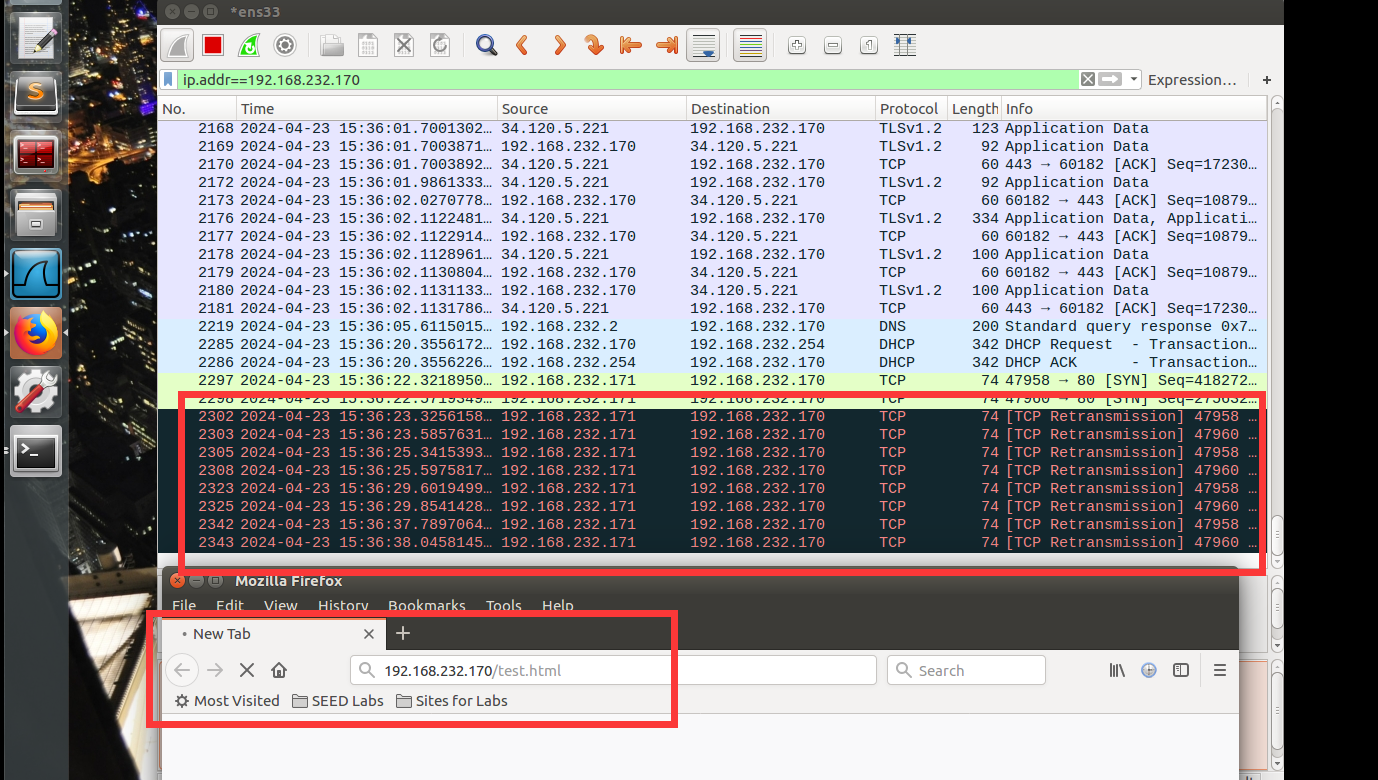


1. SEEDA和SEEDB下都再次访问test.html页面，看防火墙是否起作用：

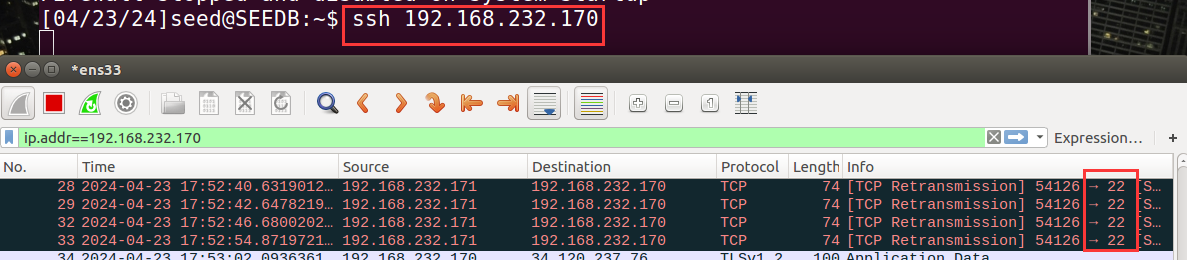
SEEDA访问，可以正常显示：



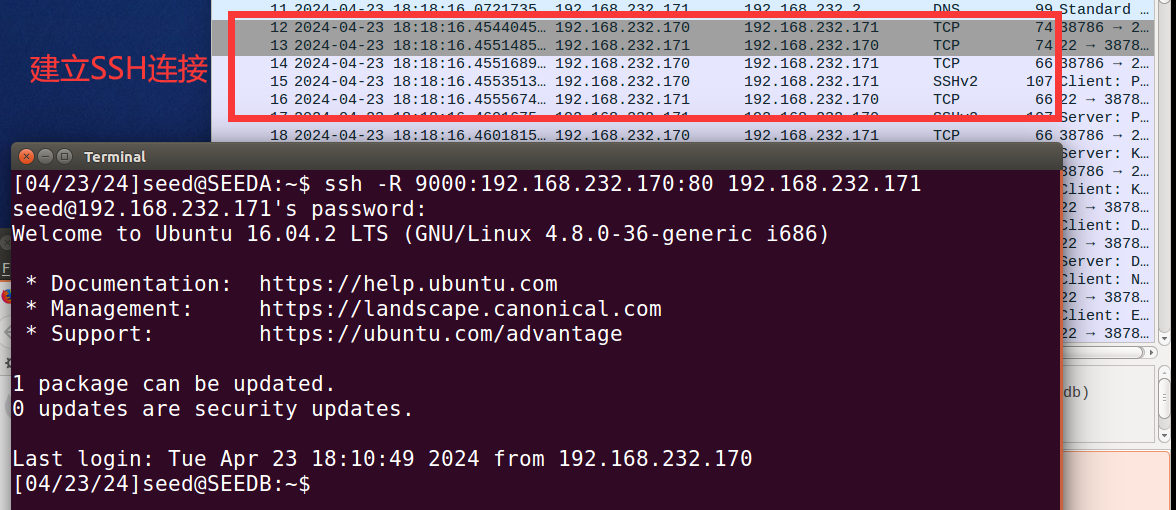
SEEDB访问，同时抓包。由于端口80被阻止，Wireshark显示从SEEDB发起的TCP连接尝试，但没有SEEDA的响应。可以看到多次的TCP重试（SYN包）尝试，最终无响应导致超时。若在SEEDA抓包，因为被防火墙过滤了，因此没显示任何包。



1. 由于SEEDA的22端口被防火墙过滤，所以，SEEDB也无法通过ssh访问SEEDA。观察wireshark，可见和访问网页一样，可以看到多次的TCP重试（SYN包）尝试，最终无响应导致超时：



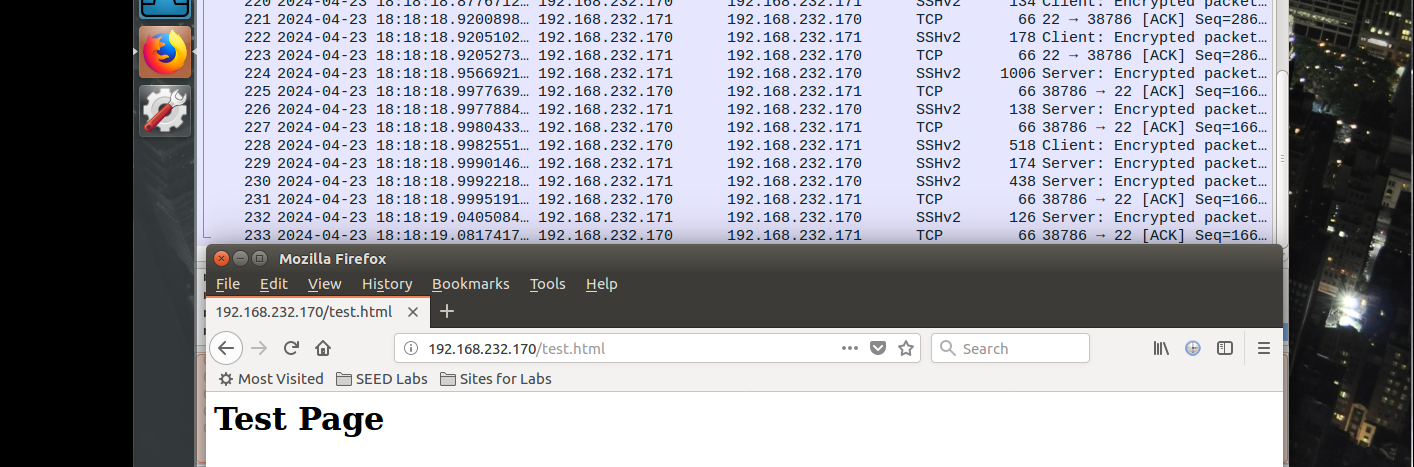
1. SEEDA下建立反向隧道ssh–R 9000: SEEDA\_ip:80 SEEDB\_ip



1. 若建立成功，SEEDB上浏览器通过9000端口访问test页面：

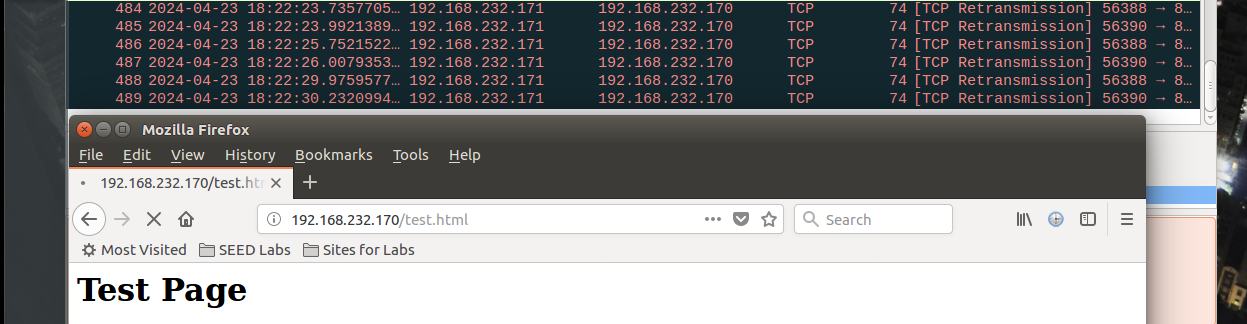
在端口9000上建立的SSH反向隧道允许流量绕过直接的端口80封锁。Wireshar显示TCP连接成功建立在端口9000，并且可以看到数据在隧道内部转发。

此时，TCP连接的特征将是SSH，这意味着HTTP请求和响应都在加密的 SSH隧道内，Wireshark无法直接解析HTTP数据，只能看到加密的TCP数据流。



1. SEEDA下退出反向隧道，再次查看test页面，比较建立和关闭反向隧道的影响:

退出反向隧道后，观察wireshark，可以看到多次的TCP重试（SYN包）尝试，最终无响应导致超时。



1. **实验思考及心得**

通过这次实验，我对防火墙技术有了更深入的理解和实践经验。实验让我熟悉了如何使用并配置UFW和iptables两种不同的防火墙工具，以及它们在实际网络环境中如何有效地控制数据流和访问权限。通过实验，我体会到防火墙在网络安全中的重要角色。防火墙不仅可以阻止未授权的访问，还能有效地管理和控制内部网络的出入流量。这是确保网络环境安全的关键工具之一。通过设置SSH 隧道来绕过防火墙的部分，让我学到了如何利用隧道技术在复杂的网络环境中保持数据传输的安全性和可访问性。

总之，这次实验不仅加深了我对防火墙配置和网络安全基础的理解，也增强了我在真实网络环境中应用这些知识的能力。我期待在未来的学习和工作中，将这些知识和技能运用到更广泛的场景中。