**实验五 Linux Firewall Exploration Lab**

**Firewall Evasion Lab: Bypassing Firewalls using VPN**

**实验报告**

57117202 朱惟璇

Lab1

**1 Task 1: Using Firewall**

Linux有一个称为iptables的工具，它实际上是一个防火墙。它有一个很好的前端程序ufw。在此任务中，目标是使用ufw设置一些防火墙策略，并在策略生效后观察系统的行为。需要至少设置两个VM，一个称为Machine A，另一个称为MachineB，在Machine A上运行防火墙。基本上使用ufw作为个人防火墙。（可选）如果有更多的VM，则可以在路由器上设置防火墙，这样它就可以保护网络，而不仅仅是一台计算机。设置两个虚拟机之后，应执行以下任务：

•阻止A对机器B进行telnet。

•阻止B对机器A进行telnet。

•阻止A访问外部网站。可以选择要阻止的任何网站，但是请记住，某些Web服务器具有多个IP地址。

可以通过键入“man ufw”找到ufw的手册或在线搜索。使用起来非常简单。请记住，默认情况下未启用防火墙，因此需要运行命令以专门启用它。在附录A中列出了一些常用命令。

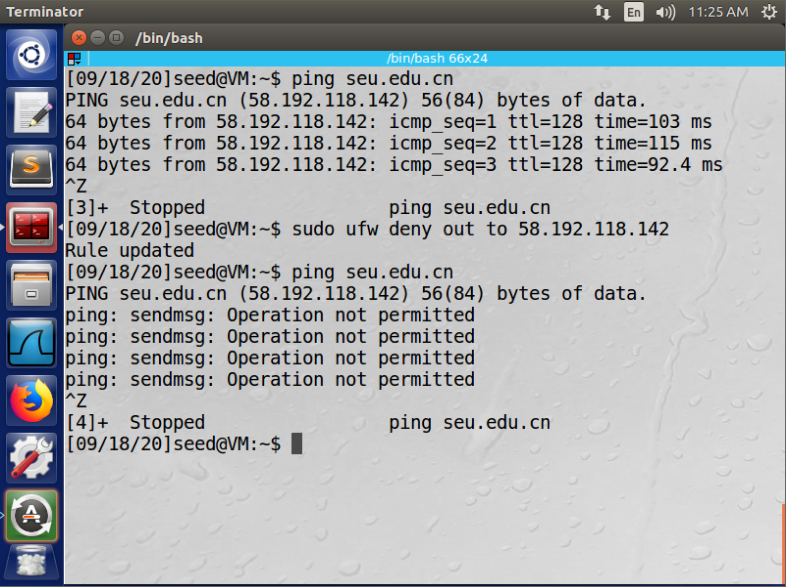
在开始任务之前，请转到默认策略文件/etc/default/ufw。查找以下条目，并将规则从DROP更改为ACCEPT；否则，默认情况下将丢弃所有传入流量。

# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if

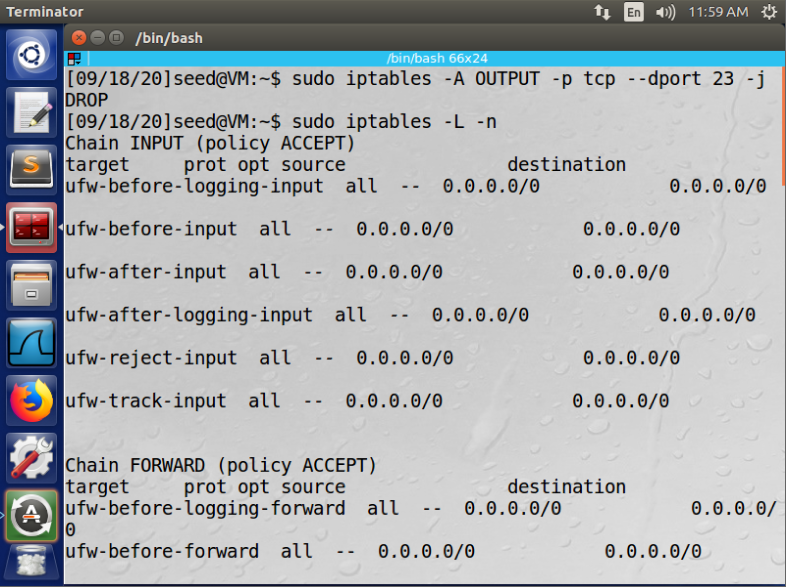
# you change this you will most likely want to adjust your rules.

DEFAULT\_INPUT\_POLICY="DROP"

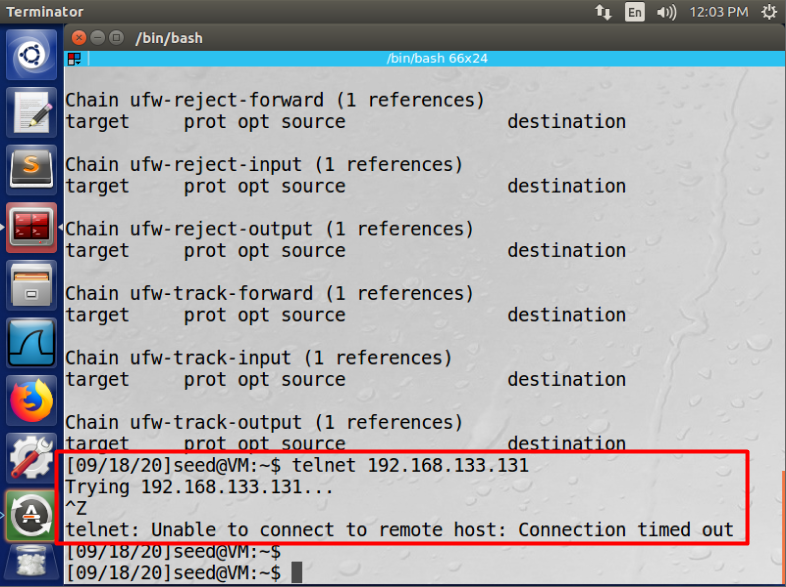
设置防火墙，使得本来可以ping seu.edu.cn的虚拟机无法联通：



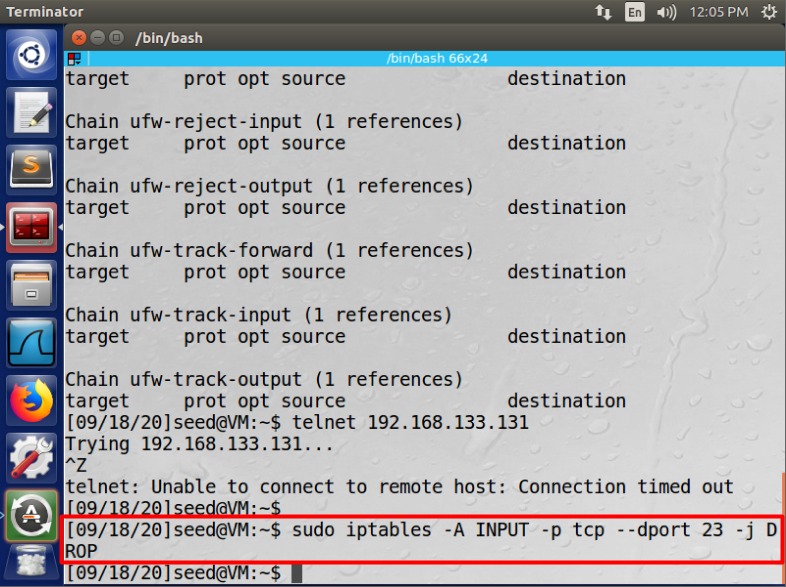
设置防火墙，阻止A对B发起连接：

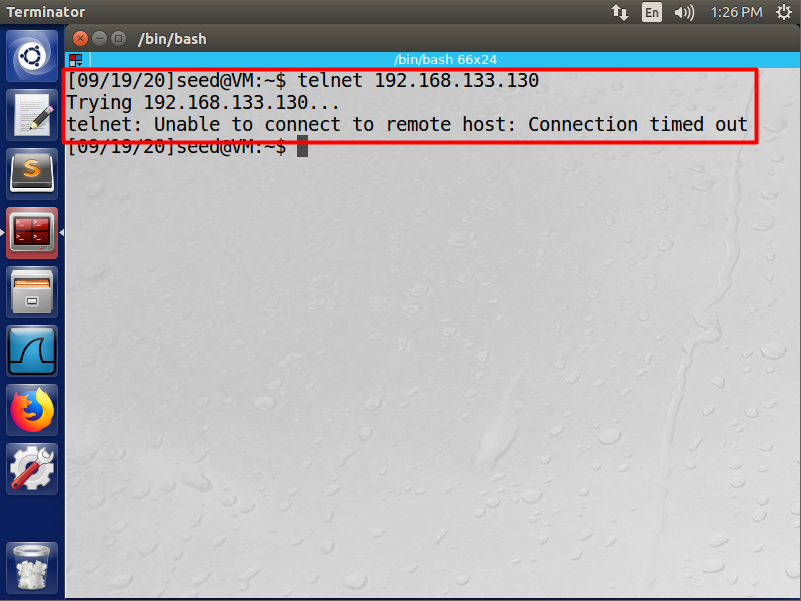


结果如图，VM A连接VM B失败：



B连接A同理，连接失败：





**2 Task 2: Implementing a Simple Firewall**

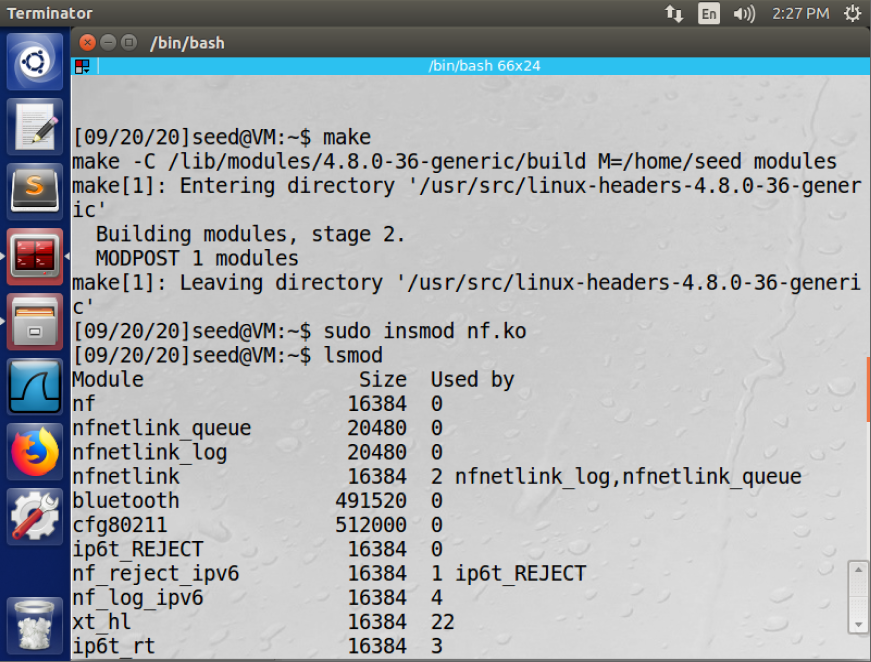
在上一个任务中使用的防火墙是数据包筛选类型的防火墙。此类防火墙的主要部分是过滤部分，该部分检查每个传入和传出的数据包，并执行管理员设置的防火墙策略。由于包处理是在内核内完成的，因此过滤也必须在内核内完成。因此，实现这种防火墙似乎需要修改Linux内核。过去，这必须通过修改和重建内核来完成。现代的Linux操作系统提供了几种新的机制来促进数据包的操作，而无需重建内核映像。这两种机制是可加载内核模块（LKM）和Netfilter。

LKM允许在运行时向内核添加一个新模块。这个新模块能够扩展内核的功能，而无需重建内核甚至不重新启动计算机。防火墙的数据包过滤部分可以实现为LKM。但是，这还不够。为了使过滤模块能够阻止传入/传出的数据包，必须将模块插入到数据包处理路径中。在将Netfilter引入Linux之前，过去很难做到这一点。

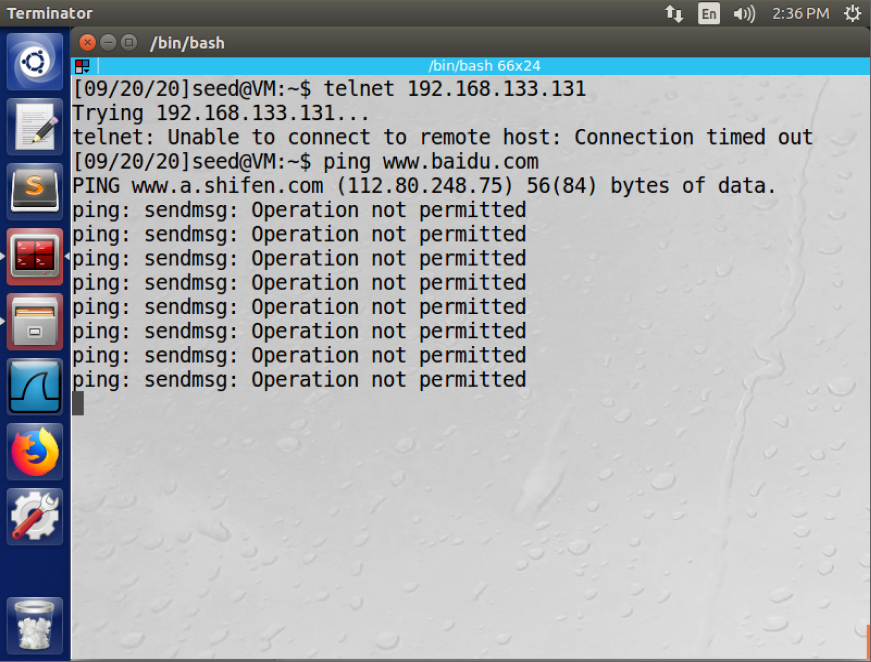
Netfilter旨在帮助授权用户处理数据包。Netfilter通过在Linux内核中实现许多挂钩来实现此目标。这些挂钩被插入到各个位置，包括数据包的传入和传出路径。如果要处理传入的数据包，只需要将自己的程序（在LKM内）连接到相应的钩子即可。一旦传入的数据包到达，程序将被调用。程序可以决定是否应阻止此数据包；此外还可以在程序中修改数据包。

在此任务中，需要使用LKM和Netfilter来实现包过滤模块。该模块将从数据结构中获取防火墙策略，并使用该策略来决定是否应阻止数据包。为了更轻松，可以专注于防火墙的核心过滤部分，允许在程序中对防火墙策略进行硬编码。应该至少支持五个不同的规则，包括上一个任务中指定的规则。有关如何使用Netfilter的指南，请参见第3节。此外，SEED书籍的第14章（第14.4节）提供了有关Netfilter的更详细的说明。

如图，配置linux内核，并重新安装，安装后显示的Netfilter配置：



配置完成后过滤包即建成，试图让VMA和VM B连接：



如图所示连接失败，说明配置成功，过滤模块已应用。

**3 Task 3: Evading Egress Filtering**

许多公司和学校都实施出口过滤，这阻止了网络内部的用户访问某些网站或Internet服务。它们确实允许用户访问其他网站。在许多情况下，这种类型的防火墙会检查传出数据包中的目标IP地址和端口号。如果数据包符合限制，它将被丢弃。由于性能原因，它们通常不进行深入的数据包检查（即，调查数据包的数据部分）。在此任务中，将说明如何使用隧道机制绕过此类出口过滤。建立隧道的方法很多，在此任务中仅关注SSH隧道。

需要两个VM A和B来完成此任务（三个会更好）。机器A在防火墙后面（即在公司或学校的网络内部）运行，而机器B在防火墙外部。通常，有一台运行防火墙的专用计算机，但是在此任务中，为了方便起见，可以在计算机A上运行防火墙。可以使用在上一个任务中实现的防火墙程序，也可以直接使用ufw。需要设置以下两个防火墙规则：

•**阻止所有传到外部telnet服务器的流量。**实际上，被阻止的服务器通常是游戏服务器或其他可能影响员工工作效率的服务器。在此任务中，使用telnet服务器进行演示。可以在机器B上运行telnet服务器。如果有第三个VM，机器C，则可以在机器C上运行telnet服务器。

•**阻止所有去往www.facebook.com的出站流量，**以使员工（或学童）在工作/上学时间不会分心。社交网站通常被公司和学校阻止。设置防火墙后，启动Firefox浏览器，并尝试连接到Facebook，并报告发生的情况。如果在使用此浏览器之前已经访问过Facebook，则需要使用Firefox的菜单清除所有缓存：Edit -> Preferences -> Privacy& Security (left pane) -> Clear History (Button on right)；否则，可能会显示缓存的页面。如果一切设置正确，则无法看到Facebook页面。应该注意的是，Facebook有许多IP地址，它可以随着时间而改变。请记住使用ping或dig命令检查地址是否仍然相同。如果地址已更改，则需要更新防火墙规则。也可以选择具有静态IP地址的网站，而不是使用Facebook。例如，大多数大学的网络服务器使用静态IP地址（例如www.syr.edu）；出于演示目的，可以尝试阻止这些IP，而不是Facebook。

除了设置防火墙规则，还需要以下命令来管理防火墙：

$ sudo ufw enable // this will enable the firewall.

$ sudo ufw disable // this will disable the firewall.

$ sudo ufw status numbered // this will display the firewall rules.

$ sudo ufw delete 3 // this will delete the 3rd rule.

**任务3.a：**通过防火墙Telnet到计算机B要绕过防火墙，可以在计算机A和B之间建立SSH隧道，因此所有telnet通信都将通过该隧道（加密），从而避免了检查。以下命令在本地主机（端口8000）和计算机B（使用默认端口22）之间建立SSH隧道；当数据包从B的末端流出时，它将被转发到计算机C的端口23（telnet端口）。如果只有两个VM，则可以将一个VM用于机器B和机器C。

$ ssh -L 8000:Machine\_C\_IP:23 seed@Machine\_B\_IP

// We can now telnet to Machine C via the tunnel:

$ telnet localhost 8000

当远程登录到本地主机的端口8000时，SSH会将所有TCP数据包从本地主机8000上的隧道的一端传输到计算机B上隧道的另一端；从那里，数据包将被转发到计算机C：23。来自计算机C的答复将采用相反的路径，并最终到达telnet客户端。本质上能够远程登录到计算机C。请描述观察并解释如何绕过出口过滤。使用Wireshark来查看导线上到底发生了什么。

**任务3.b：**使用SSH隧道连接到Facebook。为了实现此目标，可以使用类似于任务3.a中的方法，即在localhost：port和Machine B之间建立隧道，并要求B将数据包转发到Facebook。为此，可以使用以下命令来设置隧道：“ssh -L 8000：FacebookIP：80 ...”。将不使用这种方法，而是使用一种更通用的方法，称为动态端口转发，而不是像任务3.a中的静态方法。为此，仅指定本地端口号，而不指定最终目的地。机器B从隧道接收到数据包时，它将根据数据包的目标信息动态决定将数据包转发到何处。

$ ssh -D 9000 -C seed @ machine\_B

与连接localhost：9000的telnet程序类似，需要在每次需要Firefox连接到Web服务器时要求Firefox连接到localhost：9000，这样流量才能通过SSH隧道。为此可以告诉Firefox使用localhost：9000作为其代理。为了支持动态端口转发，需要一种称为SOCKS代理的特殊类型的代理，大多数浏览器都支持该代理。要在Firefox中设置代理，请转到菜单栏，单击k Edit -> Preferences，向下滚动至网络代理，然后单击设置按钮。完成设置后，请执行以下操作：

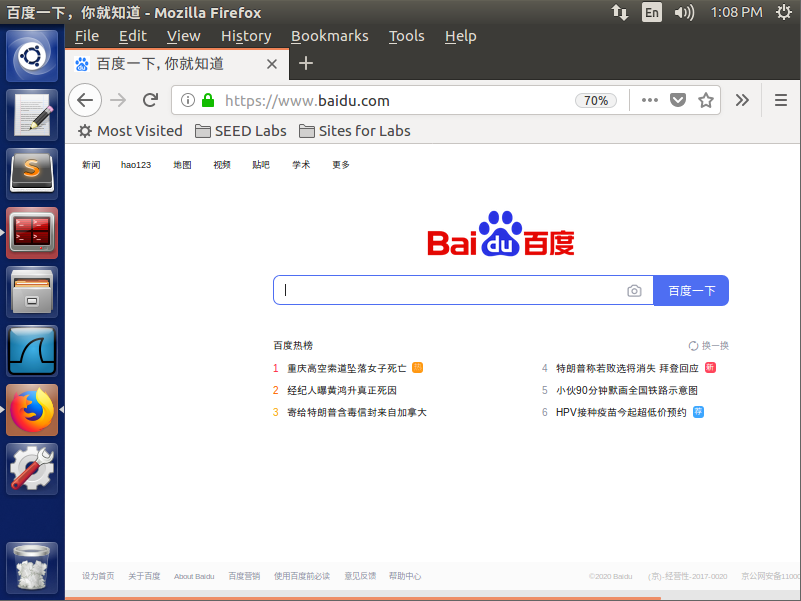
1.运行Firefox并访问Facebook页面。能看到Facebook页面吗？ 请描述观察。

2.获得facebook页面后，断开SSH隧道，清除Firefox缓存，然后重试连接。请描述观察。

3.再次建立SSH隧道并连接到Facebook。描述观察。

4.请解释所观察到的内容，特别是关于为什么SSH隧道可以帮助绕过出口筛选的原因。使用Wireshark来查看导线上到底发生了什么。请描述观察结果，并使用捕获的数据包解释它们。

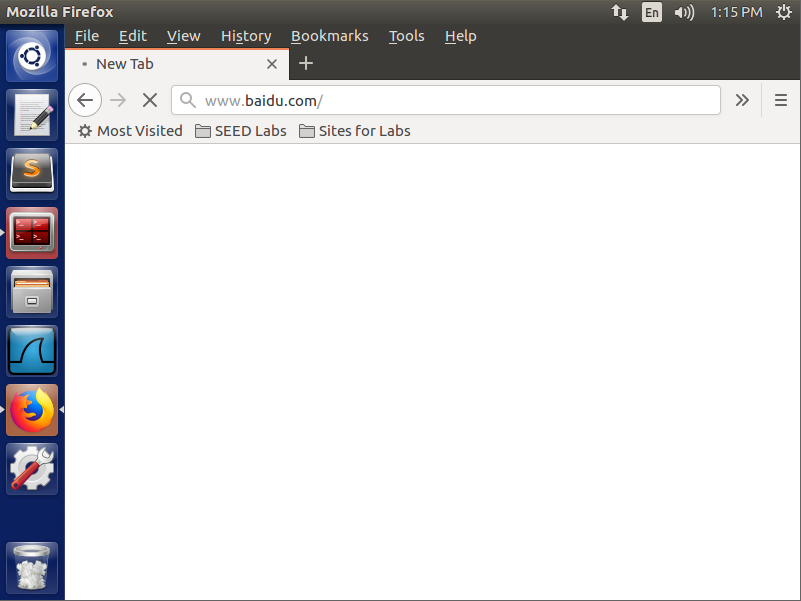
阻止所有去往www.baidu.com的出站流量，设置防火墙前，可以联通：



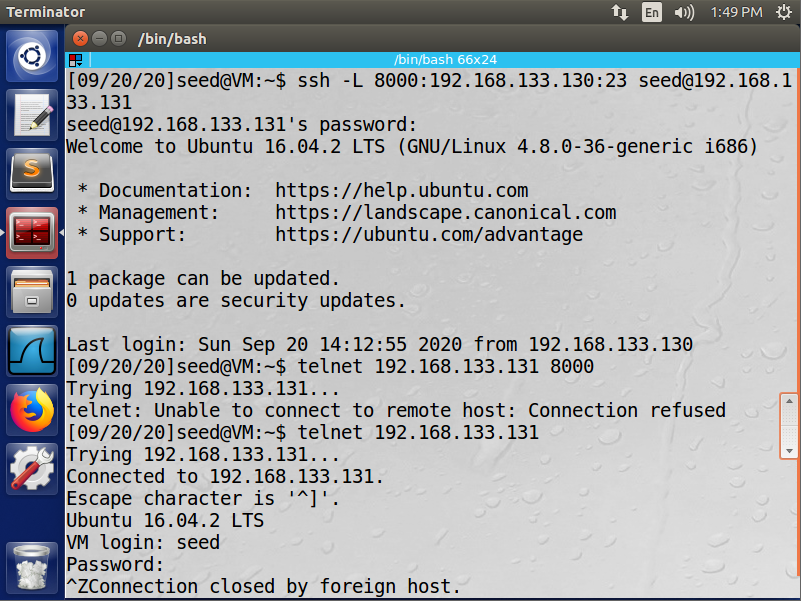
进行防火墙配置：



结果如下，无法再联通：

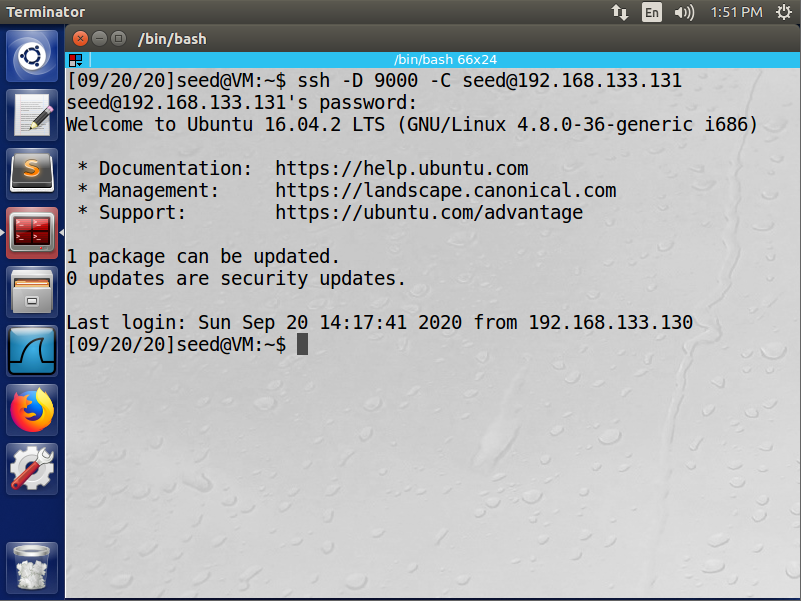


任务3.a：在计算机A和B之间建立SSH隧道，建立成功后两者可以连接，如图所示：



任务3.b：使用SSH隧道连接到Facebook。

首先建立VM A和VM B之间的SSH连接：



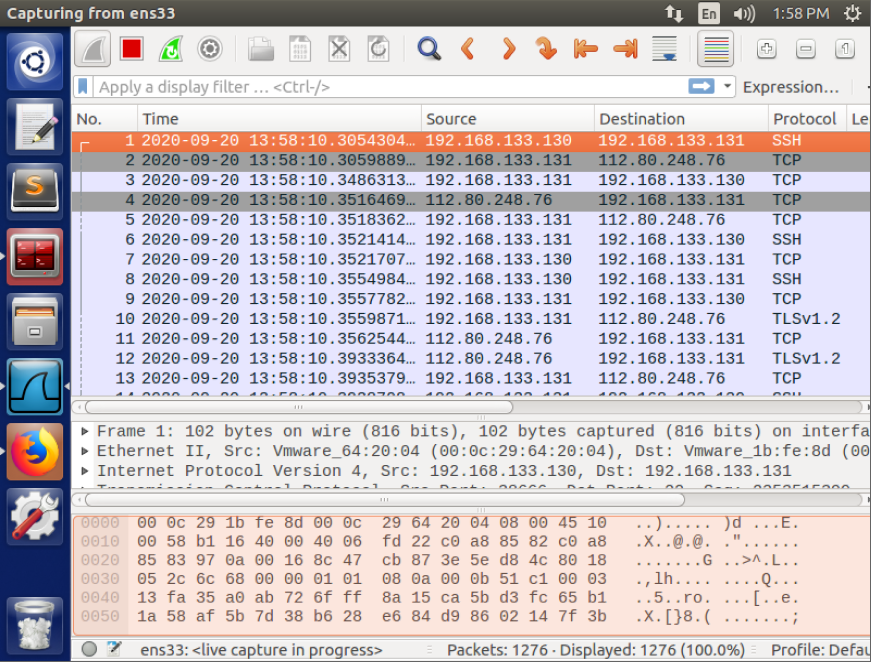
在Firefox中设置代理，设置如图所示：



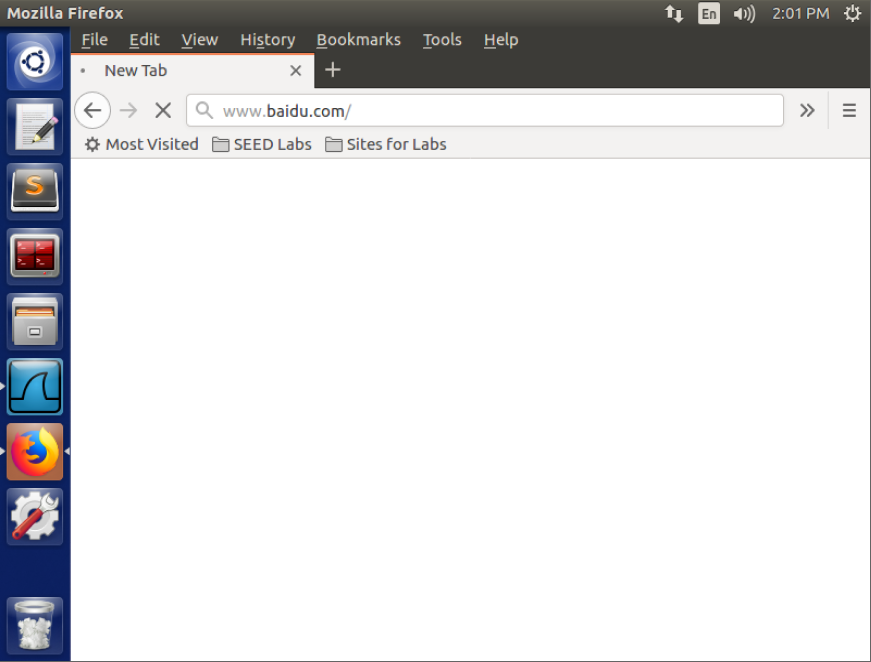
代理后可以成功访问百度：



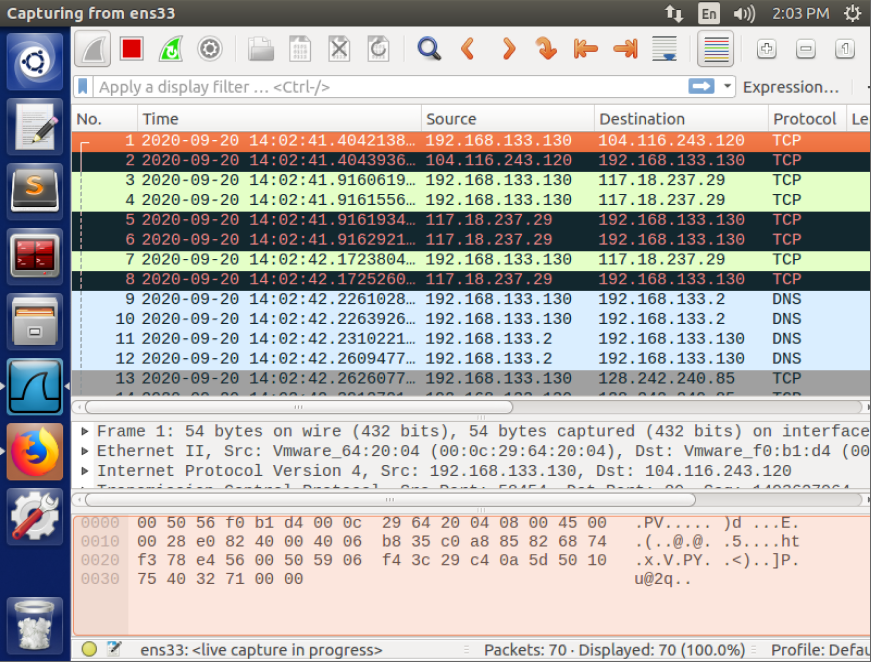
用wireshark进行抓包的结果：



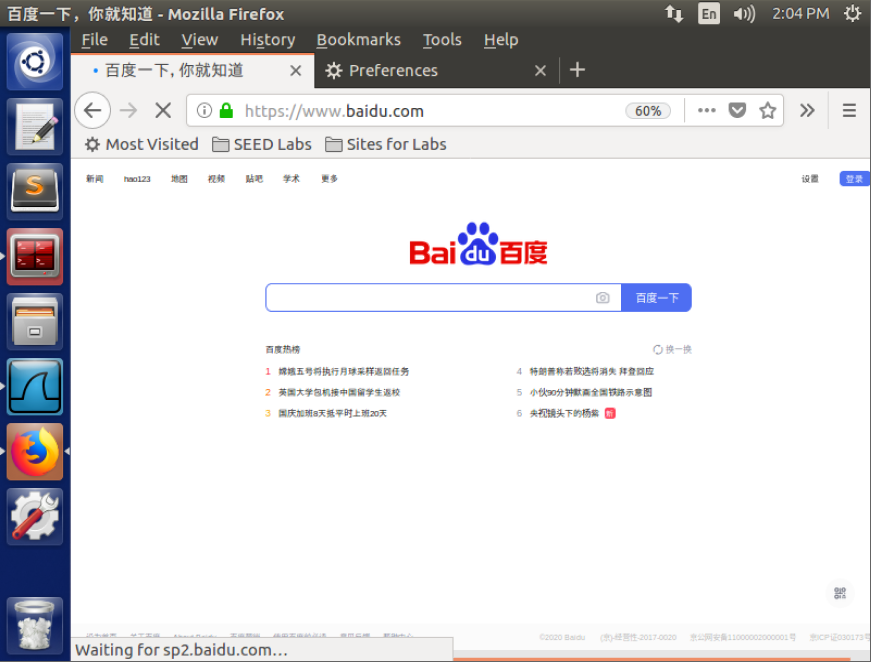
终止连接后再次访问百度发现无法访问：



可以看到协议栏已经没有SSH了：



恢复连接后即可重新访问百度，说明与百度的连接是由SSH建立的，试验成功：



**4 Task 4: Evading Ingress Filtering**

机器A在防火墙后面运行Web服务器；因此，只有内部网络中的计算机才能访问此Web服务器。正在家里工作，需要访问此内部Web服务器。没有VPN，但有SSH，SSH被认为是穷人的VPN。确实在计算机A（或防火墙后面的另一台内部计算机）上有一个帐户，但是防火墙也阻止了传入的SSH连接，因此无法SSH到内部网络上的任何计算机。否则，可以使用任务3中的相同技术来访问Web服务器。但是，防火墙不会阻止传出SSH连接，即如果要连接到外部SSH服务器，仍然可以这样做。

此任务的目的是能够从外部访问计算机A上的Web服务器。将使用两台机器来模拟设置。机器A是内部计算机，运行受保护的Web服务器；机器B是在家中的外部机器。在计算机A上，阻止计算机B访问其端口80（Web服务器）和22（SSH服务器）。需要在计算机A上建立反向SSH隧道，因此一旦回到家中，仍然可以从家中访问A上受保护的Web服务器。