# 中南大学

# TCP/IP 攻击实验
# 实验报告


学生姓名 范弘毅

学　　院 信息科学与工程学院

专业班级 信安 1401


完成时间 2016 年 11 月 20 日

# TCP/IP 攻击实验

## 1.实验描述

**【实验背景】**

由于 TCP/IP 协议是 Internet 的基础协议，所以对 TCP/IP 协议的完善和改进是非常必要的。TCP/IP 协议从开始设计时候并没有考虑到现在网络上如此多的威胁,由此导致了许多形形色色的攻击方法，一般如果是针对协议原理的攻击(尤其 DDOS)，我们将无能为力。

TCP/IP 攻击的常用原理有：

(1) 源地址欺骗(Source Address Spoofing)、IP 欺骗(IP Spoofing)和 DNS 欺骗(DNS Spoofing)；

(2) 路由选择信息协议攻击(RIP Attacks)；

(3) 源路由选择欺骗(Source Routing Spoofing) ；

(4) TCP 序列号欺骗和攻击(TCP Sequence Number Spoofing and Attack)。

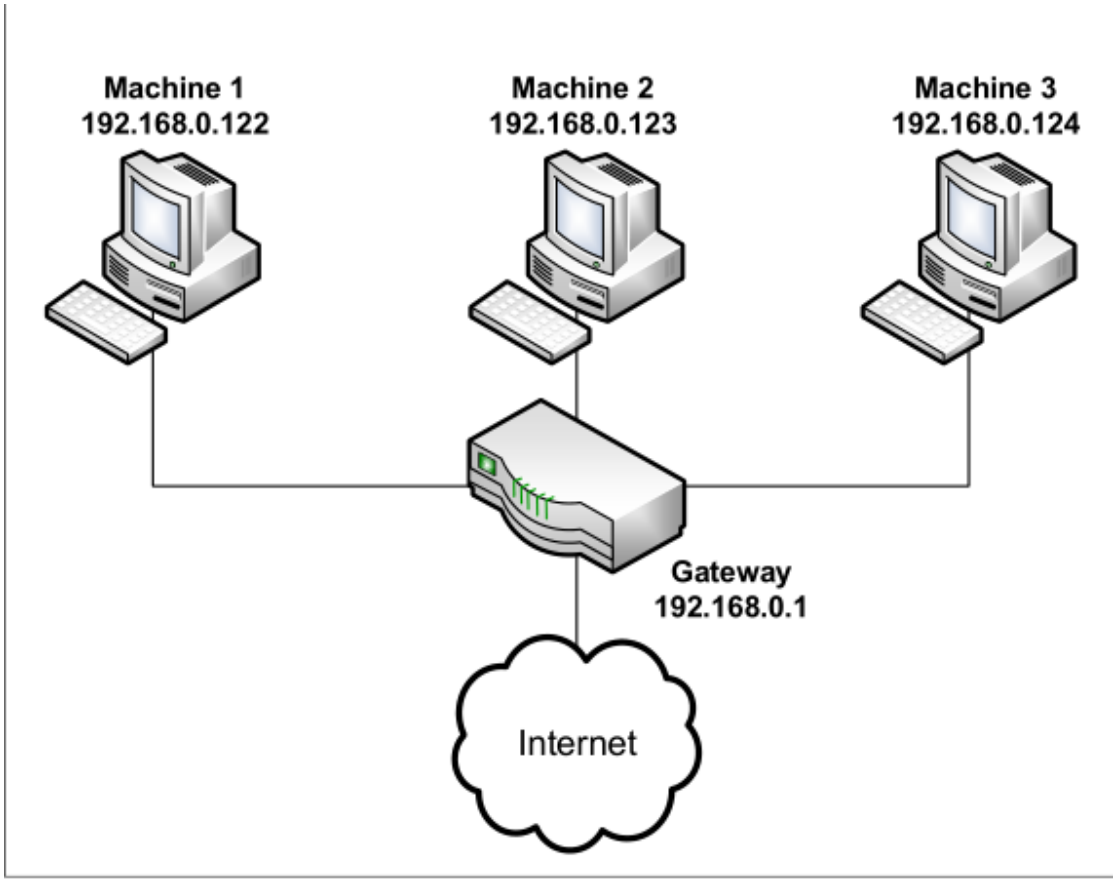
**【实验目的】**

基于 TCP/IP 协议进行攻击实验,了解 TCP/IP 协议的具体机制。

## 2.实验步骤

### 2.1 环境搭建

这里我使用三台虚拟机做实验，其中一个用于攻击；另一个用于被攻击；第三个作为观察者使用；把三台主机放在同一个 LAN 中，其配置信息参照如下所示（实际在实验过程中有所改动）：




这里我使用的是 SEED 实验室已经搭建好，并且已经安装好相关的 netwox 工具箱和 Wireshark 工具箱的 Ubuntu 系统，与此同时三台虚拟机都需要打开 FTP 和 Telnet 服务：

使用如下命令来完成上述任务

Start the ftp server

# servicevsftpd start

Start the telnet server

# serviceopenbsd-inetd start

## 2.2 实验 1：SYN 洪流攻击

**【实验背景】**

SYN 洪流攻击是 Dos 攻击的一种形式，攻击者发送许多 SYN 请求给受害者的 TCP 端口，但是攻击者没有完成三次握手的意向。攻击者或者使用虚假的 IP 地址，或者不继续过程。在这个攻击中，攻击者可以使受害者的用于半开连接的队列溢出，例如，一个完成 SYN，SYN-ACK 但没有收到最后的 ACK 回复的连接。当这个队列满了的时候，受害者不能够在进行更多的连接。

SYN 缓存策略：SYN 缓存是是对抗 SYN 洪流攻击的一种防御机制。如果机器检测到它正在被 SYN 洪流攻击，这种机制将会 kick in。

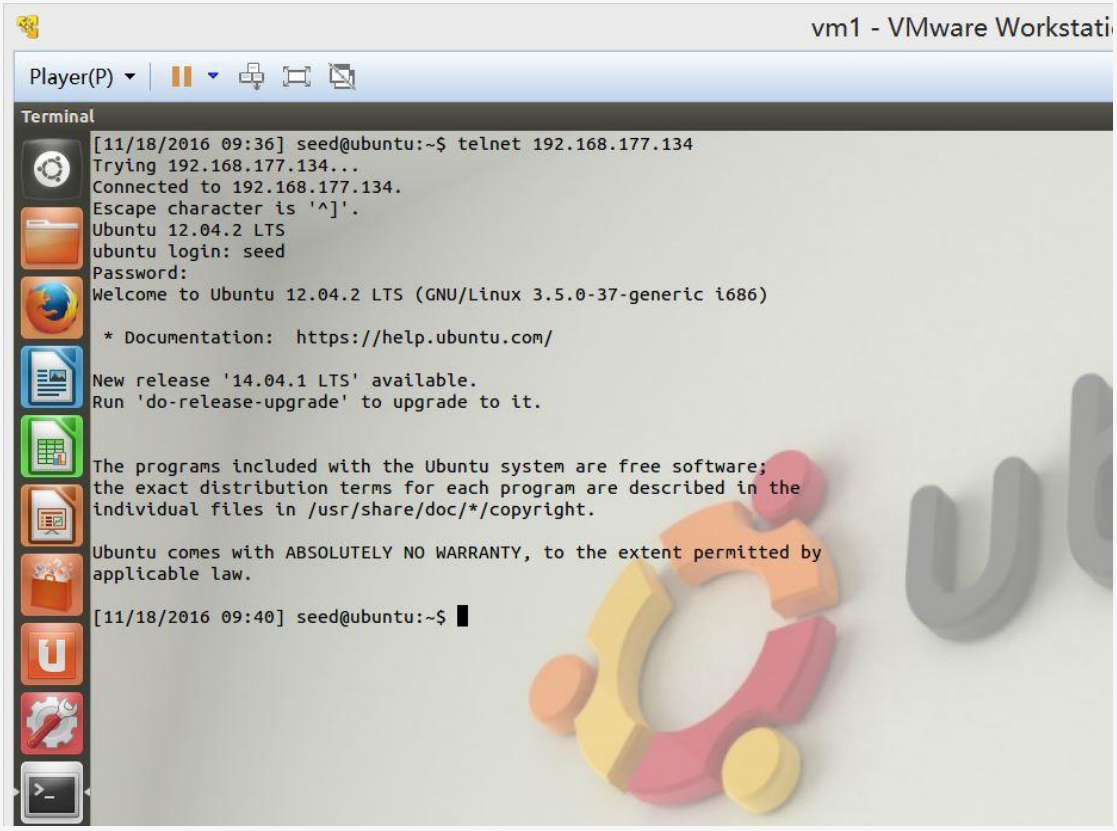
**【实验内容】**

如果一个 TCP 连接没有完成三次握手，它将被放入半开连接队列，而半开连接队列有最大长度，如果连接数量达到最大容量时，新的连接就不能够被建立。SYN 洪泛攻击就是

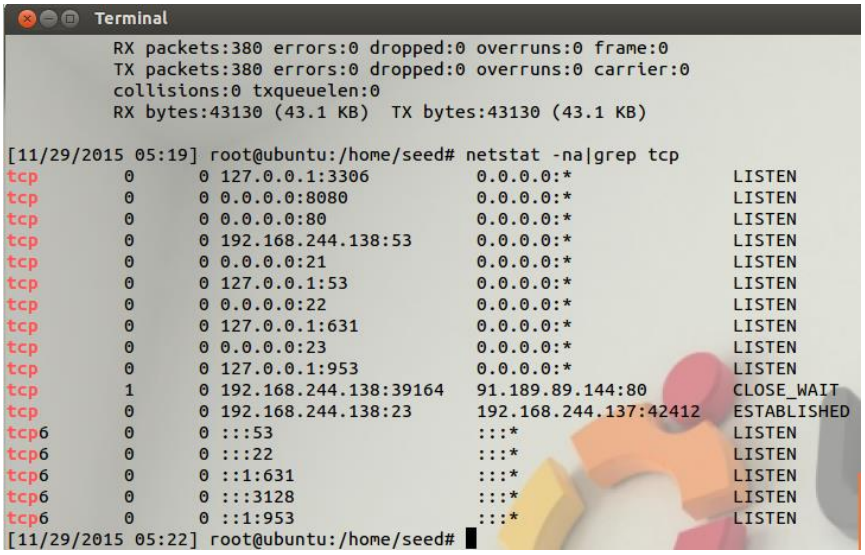
通过未完成的 TCP 请求来试图充满半开连接队列，使得正常的连接不能够被建立，达到攻击的效果。

在这个实验中，使用 telnet 服务作为攻击目标，在 23 号端口发起 SYN 洪泛攻击。

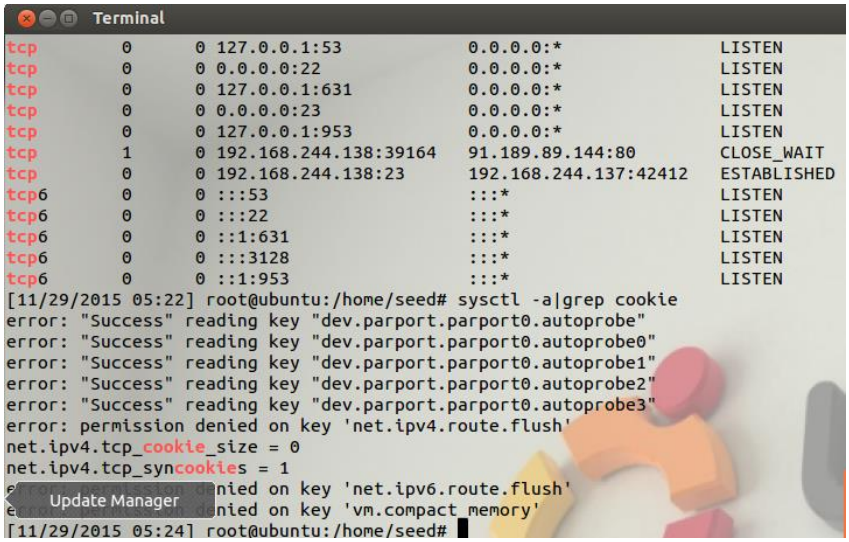
首先，尝试在主机 B 和 C 之间建立 telnet 连接，说明网络联通。主机 B 远程登录主机 C 的账户



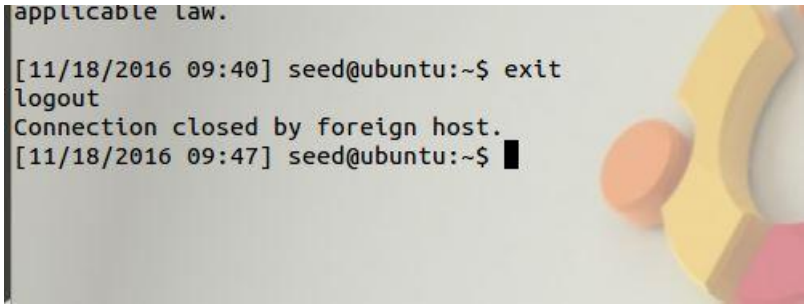

在主机 C 上，通过命令 netstat –na | grep tcp 命令查看当前的 TCP 相关端口的状态，发现 23 号端口处于联通状态

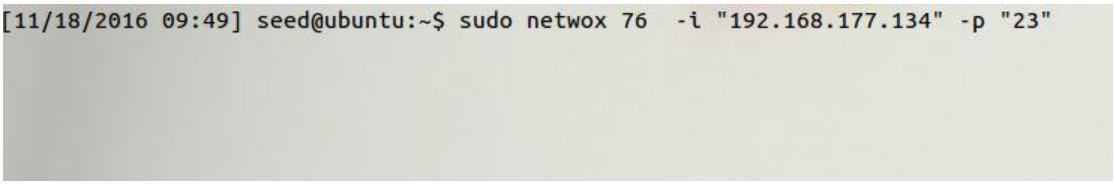



在主机 C 上查看 C 的半开连接队列的最大长度为 128，缓冲保护开启。

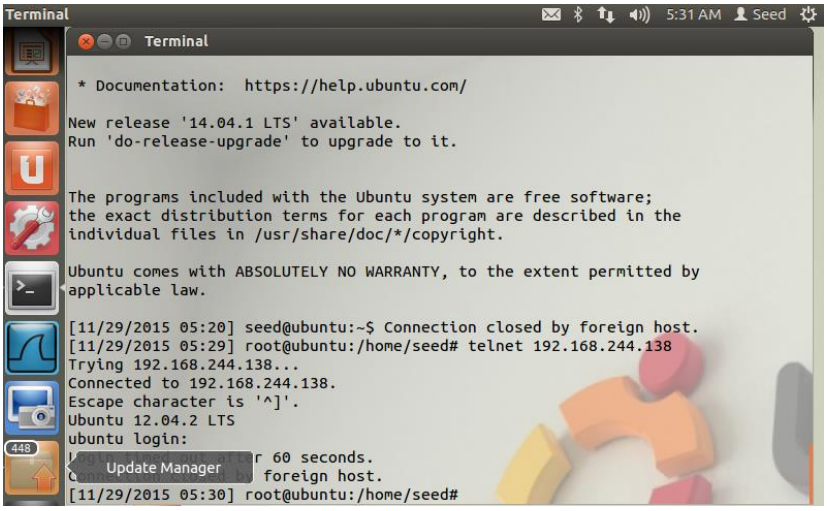



在主机 B 中使用 exit 命令断开与 C 的 telnet 连接。之后在主机 A 中使用 netwox76 号工具发动针对主机 C23 号端口的 SYN 攻击。



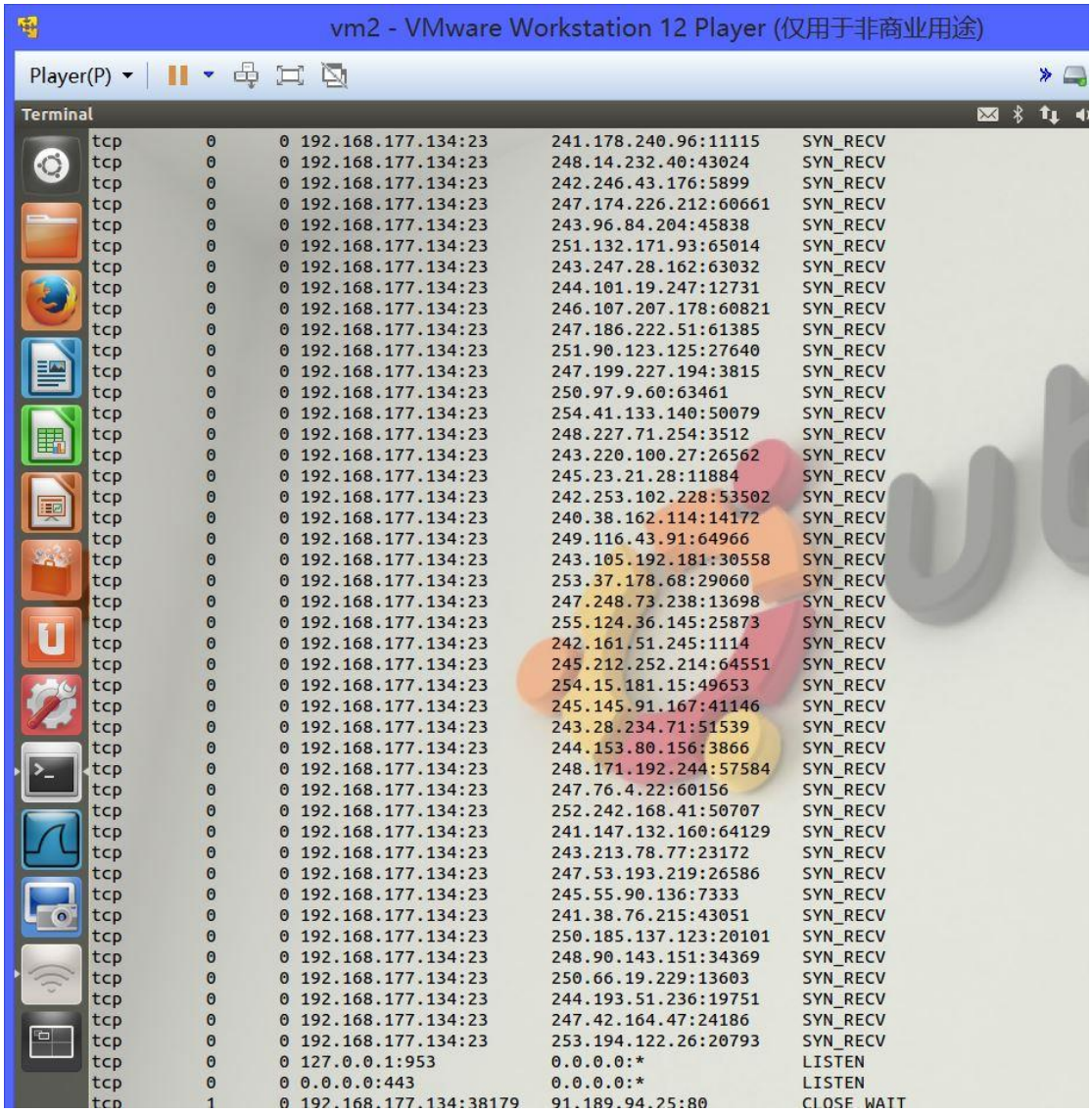




回到主机 B 中，尝试与主机 C 进行 telnet 远程连接，




从上图及实验过程可以看出，虽然连接的速度很慢，但是是可以连接上的。我在主机 B 上开启了两个终端，同时试图进行 telnet 连接。

到主机 C 中查看端口连接情况，如图 4.3.5 和图 4.3.6。发现，队列中充斥着大量半开连接，目的端口号都是 C 机的 23 号端口，但是源主机 IP 和端口却不一致，而且端口号都是不常用端口，可以判断出，这极有可能是一次 SYN 攻击。



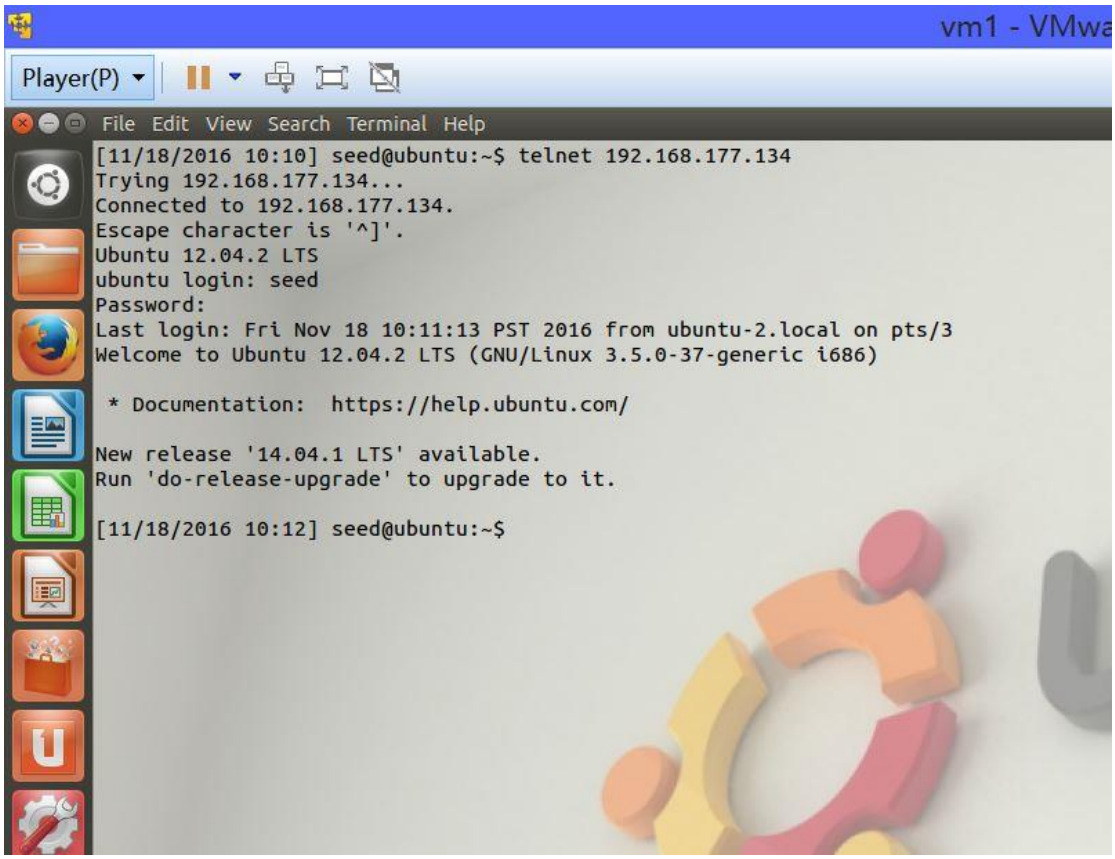

## 2.3 实验 2：在 telnet 和 ssh 连接上的 TCP RST 攻击

**【实验背景】**

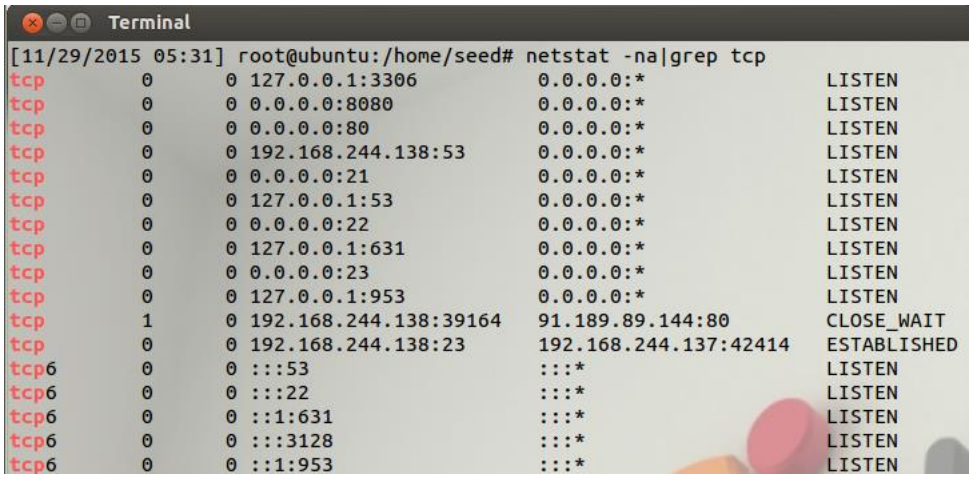
TCP RST 攻击可以终止一个在两个受害者之间已经建立的 TCP 连接。例如，如果这里有一个在 A 和 B 之间已经建立的 telnet 连接，攻击者可以伪造一个 A 发向 B 的 RST 包，打破这个存在的连接。

**【实验内容】**

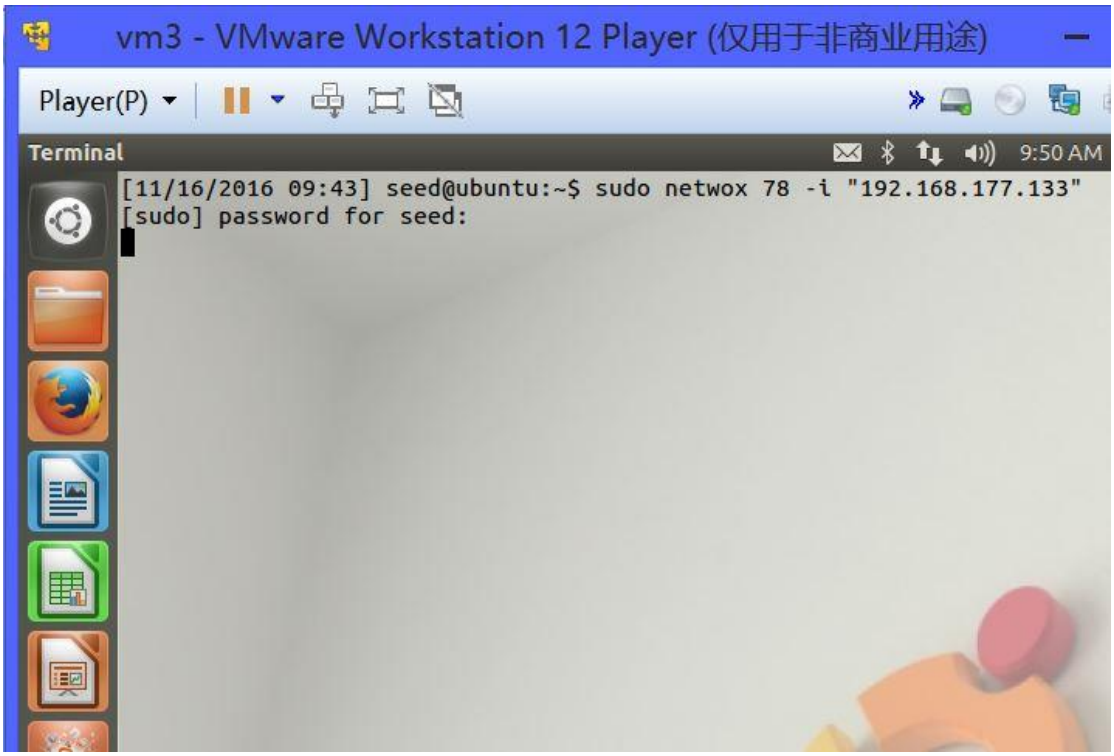
首先完成主机 B 与主机 C 的 telnet 连接，




在 C 上查看端口连接情况，如图 4.4.2，已经完成主机 B 与主机 C23 端口的连接。

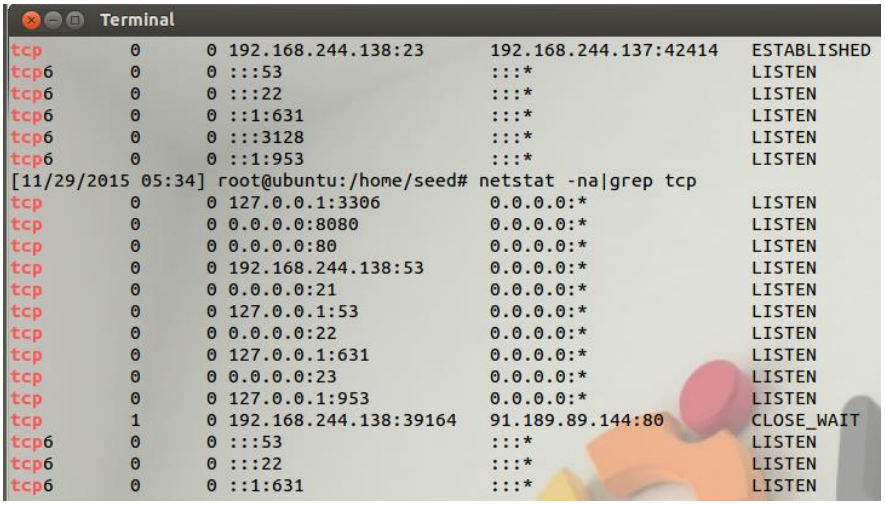



这时，在主机 A 中通过 netwox78 号工具发起针对 B 主机的 RST 攻击。

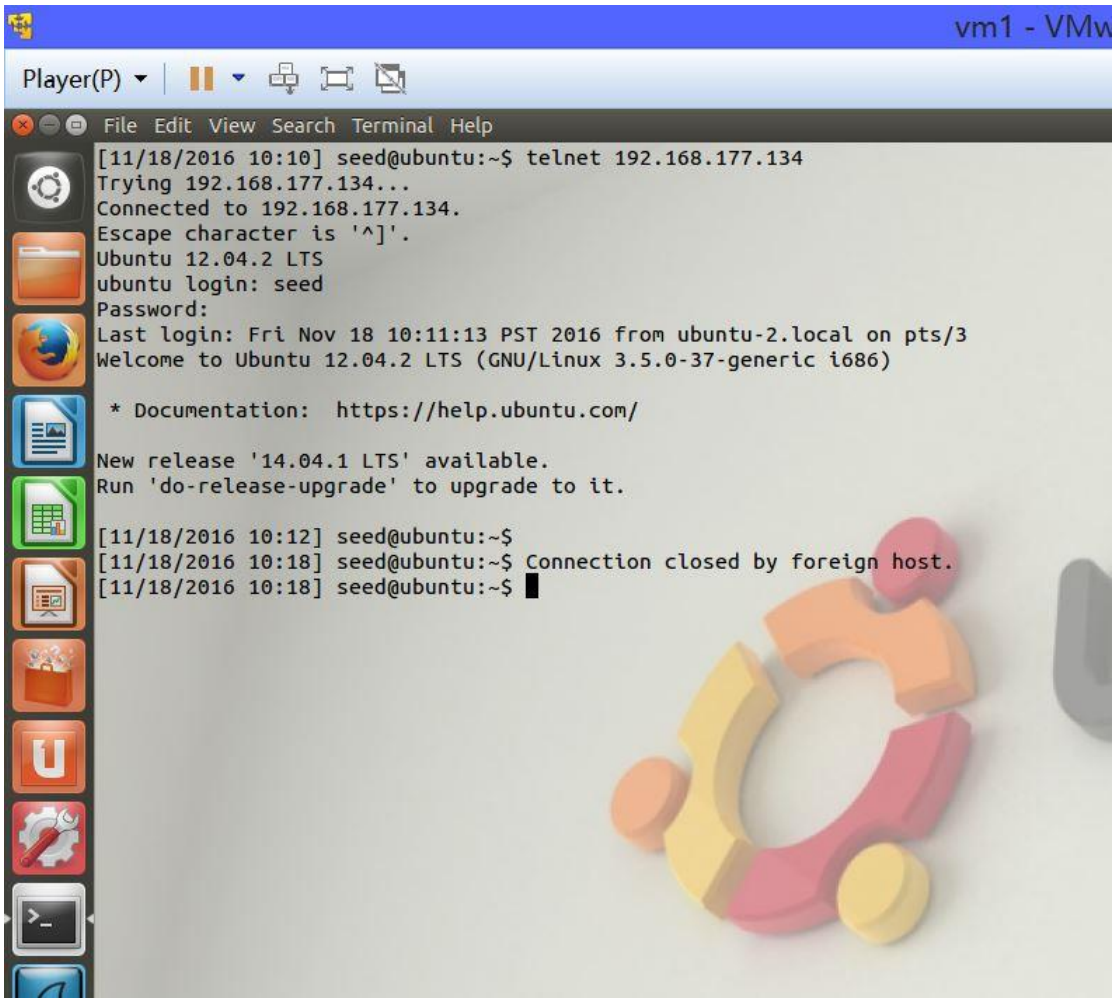



回到 B 主机中，发现没有什么变化，但是当回车之后，出现连接已经被其他主机断开，并退回到主机 B 的账户下

在主机 C 中查看此时的连接情况，如图 4.4.4。可以看出 BC 主机的 23 端口的连接已经被断开，处于监听状态。




注意，此时主机 A 的攻击并没有停止。回到主机 B 中，再次尝试连接主机 C，发现最开始是连接上了，但是还没来得及显示后续内容，连接就被中断。




# 3.总结

通过这一次内容丰富并且工作量巨大的实验，我对基于 TCP/IP 的攻击有了更加深刻甚至可以说是比较新的认识，对它们各自的机制、攻击特点、相互之间可能存在的联系以及它们差别所在等等细节问题有了新的看法、认识。这次实验，让我至少意识到了以下这样一个事实：TCP/IP 协议在设计之初仅考虑了成本和实现功能，并没有过多考虑安全因素。因此 TCP/IP 协议栈中提供了大量的起关键作用的信息和指令，但是这些信息和指令的执行缺乏认证机制，能够方便地伪造。这也就为如此之多的 TCP/IP 攻击提供了可能。