**Lab2 TCP/IP Attack Lab**

57118101 卞郡菁

准备工作：

**一、容器构建**

第一个窗口（后台运行）：

cd Desktop/Labs\_20.04/Network\ Security/TCP\ Attacks\ Lab/Labsetup

dcbuild

dcup

第二个窗口（攻击窗口）：

cd Desktop/Labs\_20.04/Network\ Security/TCP\ Attacks\ Lab/Labsetup

dockps

docksh b2

第三个窗口（victim主机A窗口，ip为10.9.0.5）

cd Desktop/Labs\_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\ Lab/Labsetup/

dockps

docksh 23

第四个窗口（主机B窗口）

cd Desktop/Labs\_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\ Lab/Labsetup/

dockps

docksh 2c

第五个窗口（主机C窗口）

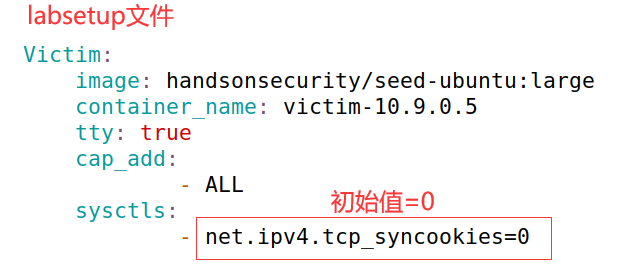
cd Desktop/Labs\_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\ Lab/Labsetup/

docksh 37

docksh a2

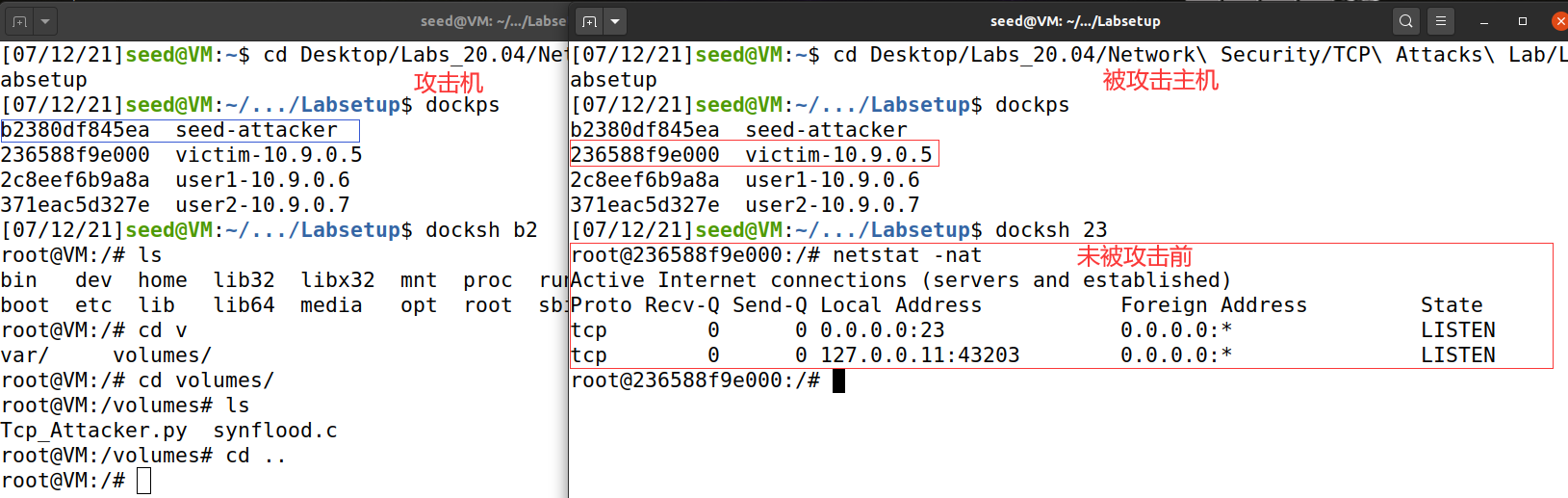
**二、配置文件**

Ubuntu本来自动开启syncookies=1故而可以抵抗泛洪攻击，为演示攻击效果，容器中默认此处为0。在labsetup文件夹中的配置文件docker-compose.yml中，

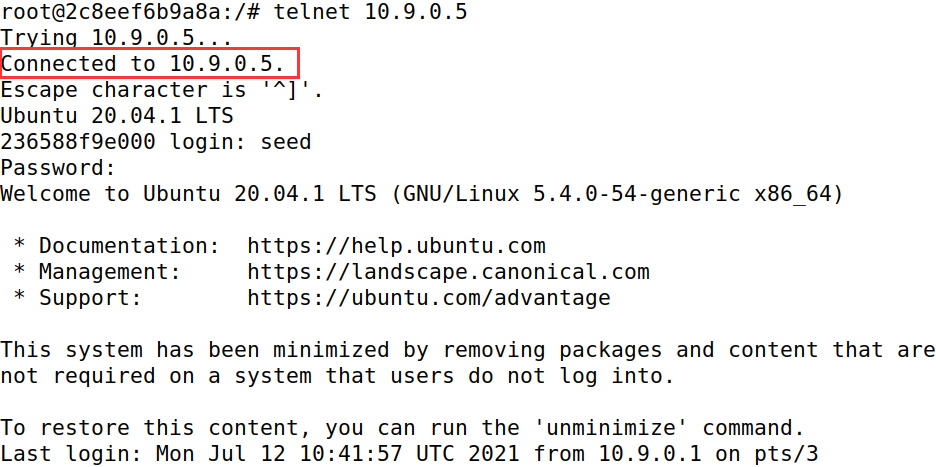


**Task 1: SYN Flooding Attack**

在未攻击前，victim主机使用netstat- nat命令查看自身tcp状态。结果表明：目前活跃的网络连接仅为两个端口Listen状态。

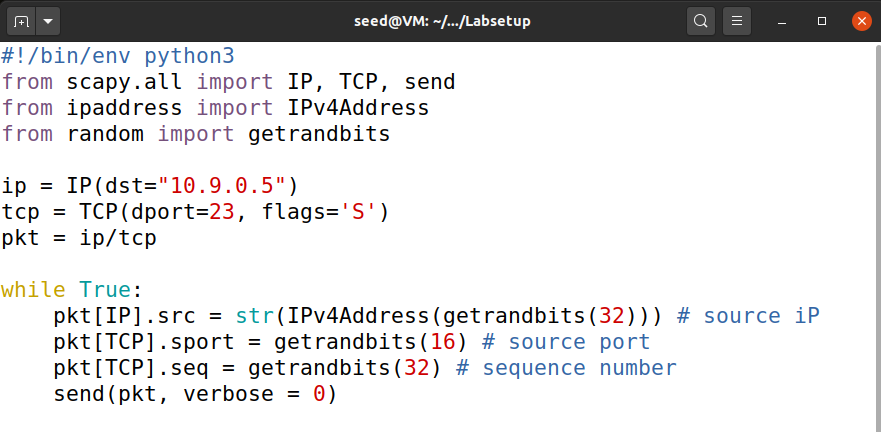


在未攻击前，使用主机B（10.9.0.6） telnet此 victim主机（10.9.0.5），发现可以正常登录。



* 1. **使用Python语言脚本攻击主机**

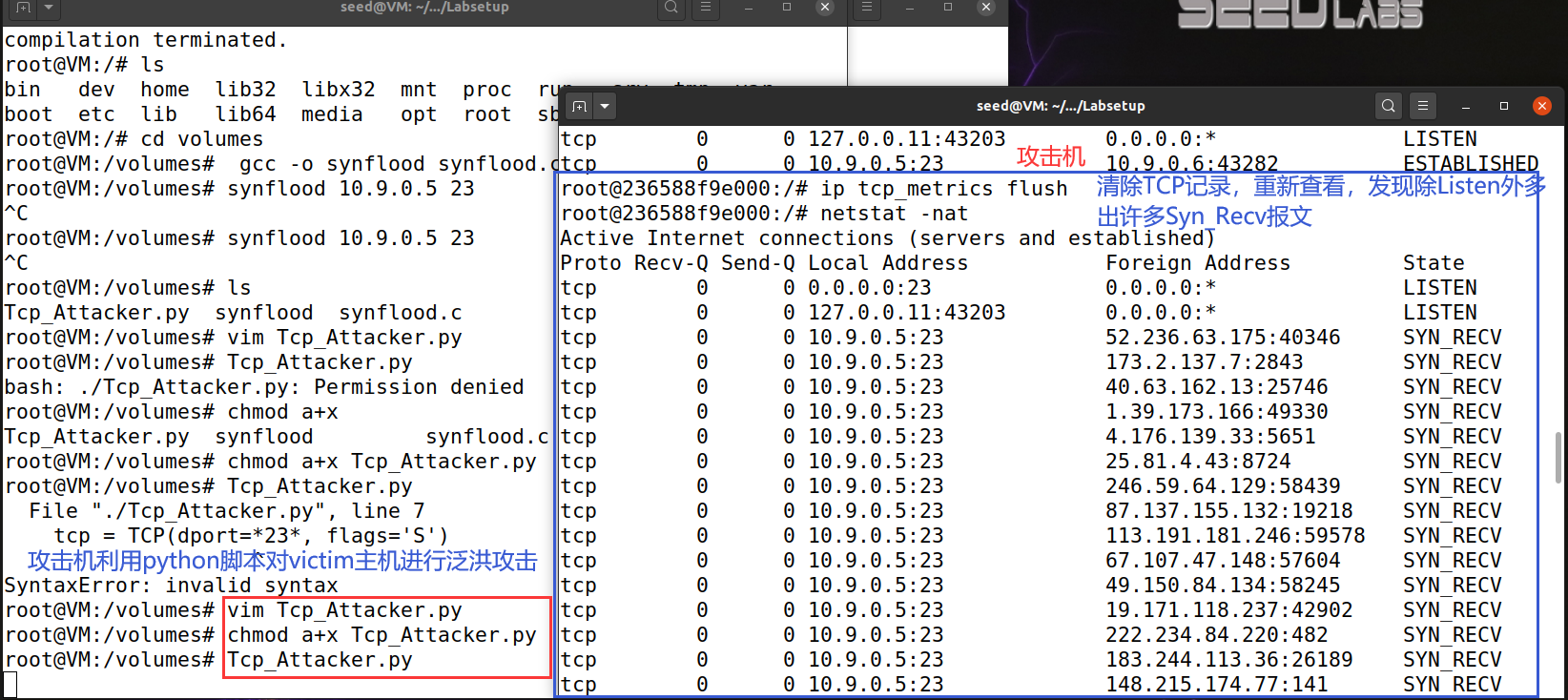
在攻击机中，编写Tcp\_Attacker.py脚本：



攻击机编译并执行Tcp\_Attacker.py程序，对victim主机（ip地址为10.9.0.5）的23号端口（telnet端口）进行SYN泛洪攻击；

Victim主机再次执行netstat -nat命令，查看TCP连接。

结果表明：除原本的两个Listen状态外，还出现了大量的Syn-Recv状态，说明victim主机已被进行Syn泛洪攻击。



在攻击机对victim主机进行泛洪攻击时，再使用主机B（ip为10.9.0.6）去远程登录victim主机。结果表明：超时，登录不上，说明其忙于泛洪攻击应答。

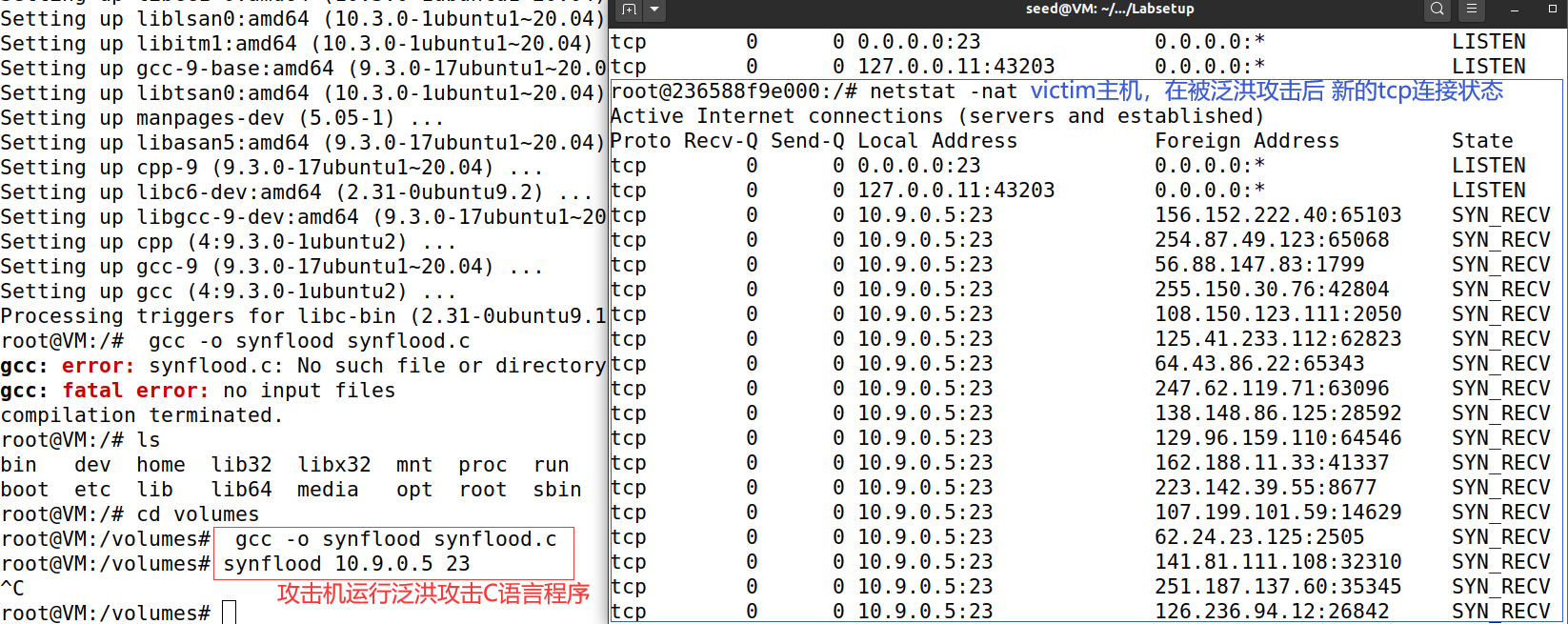


* 1. **使用C语言脚本攻击主机**

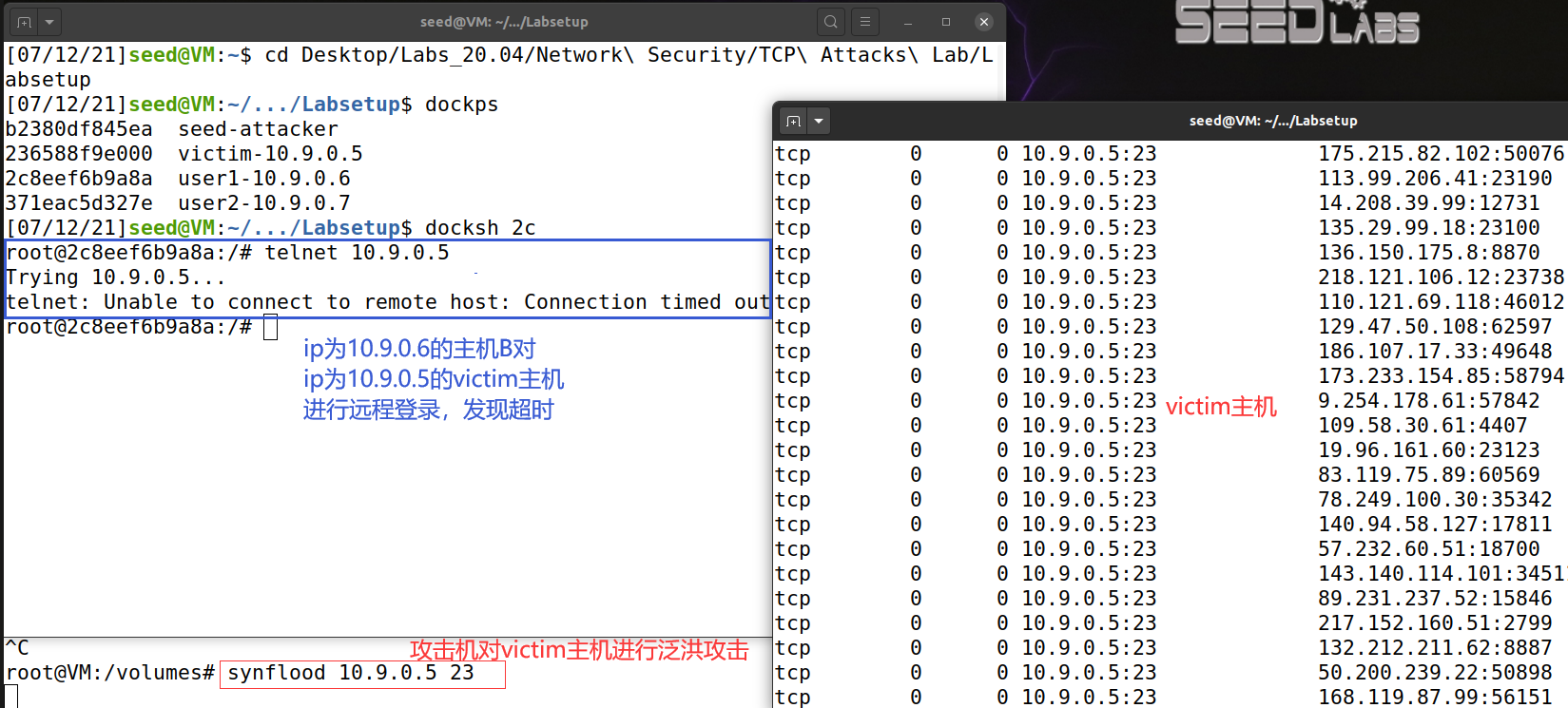
攻击机编译并执行synflood.c程序，对victim主机（ip地址为10.9.0.5）的23号端口（telnet端口）进行SYN泛洪攻击；

Victim主机再次执行netstat -nat命令，查看TCP连接。

结果表明：除原本的两个Listen状态外，还出现了大量的Syn-Recv状态，说明victim主机已被进行Syn泛洪攻击。



在攻击机对victim主机进行泛洪攻击时，再使用主机B（ip为10.9.0.6）去远程登录victim主机。结果表明：超时，登录不上，说明其忙于泛洪攻击应答。



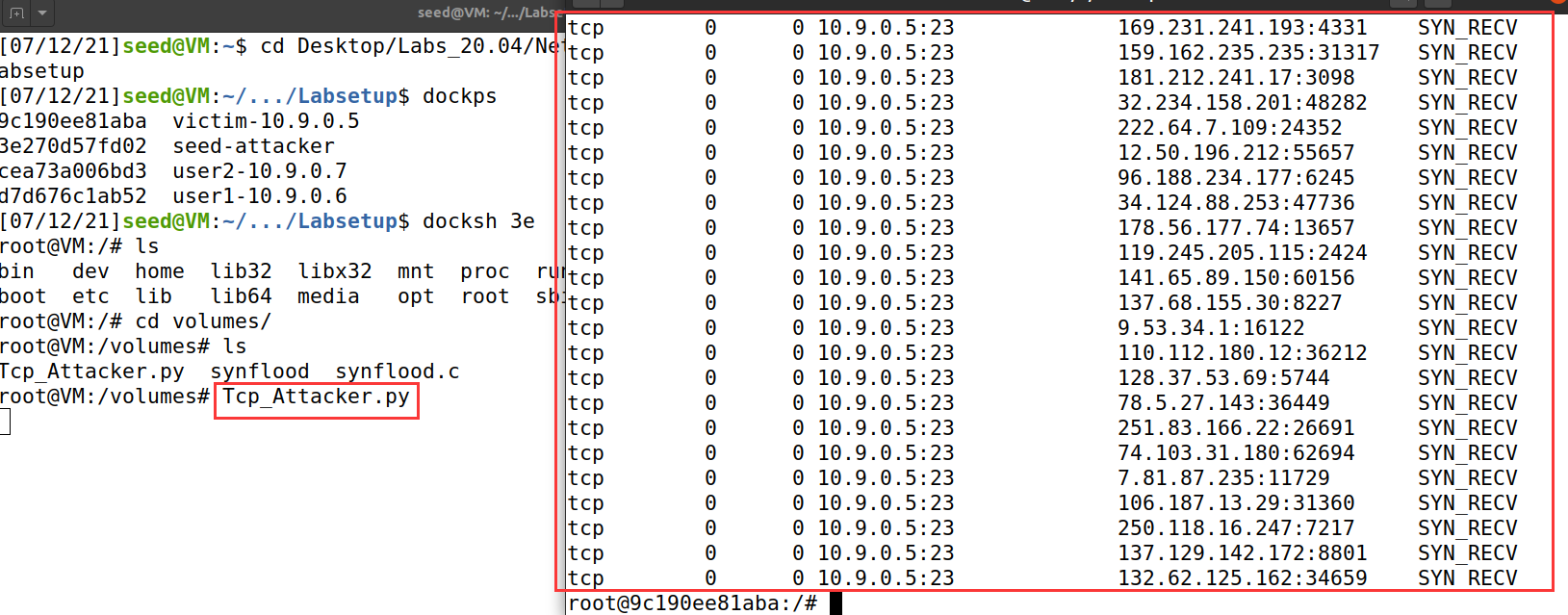
* 1. **启动SYN Cookie机制**

修改labsetup文件夹中的配置文件docker-compose.yml文件，将syncookies改为1以便自动抵御泛洪攻击。

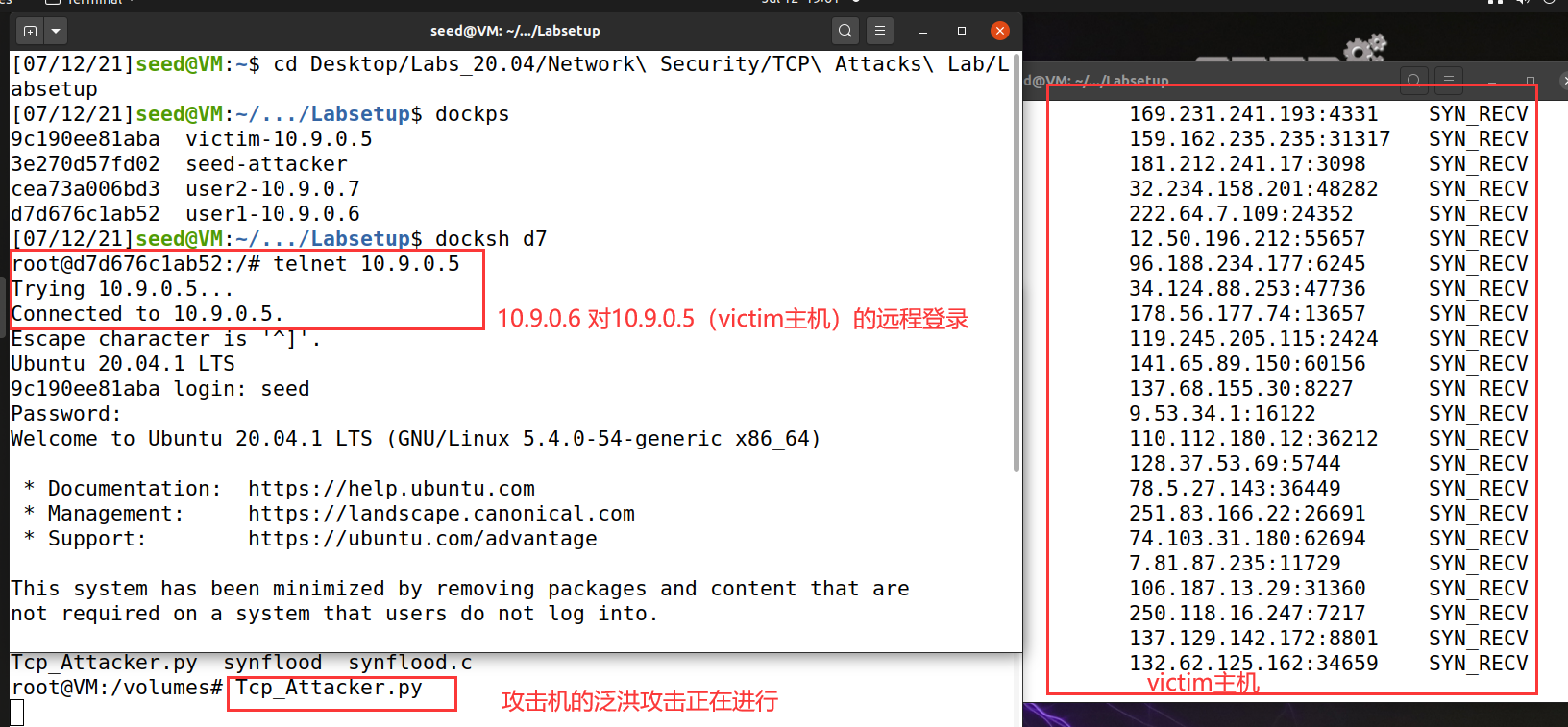


随便运行Tcp\_Attacker.py和synflood.c之一来构造泛洪攻击，再查看victim主机的TCP列表。结果表明：

1. 使用netstat -nat查看，victim主机仍然有很多SYN\_Recv状态连接。



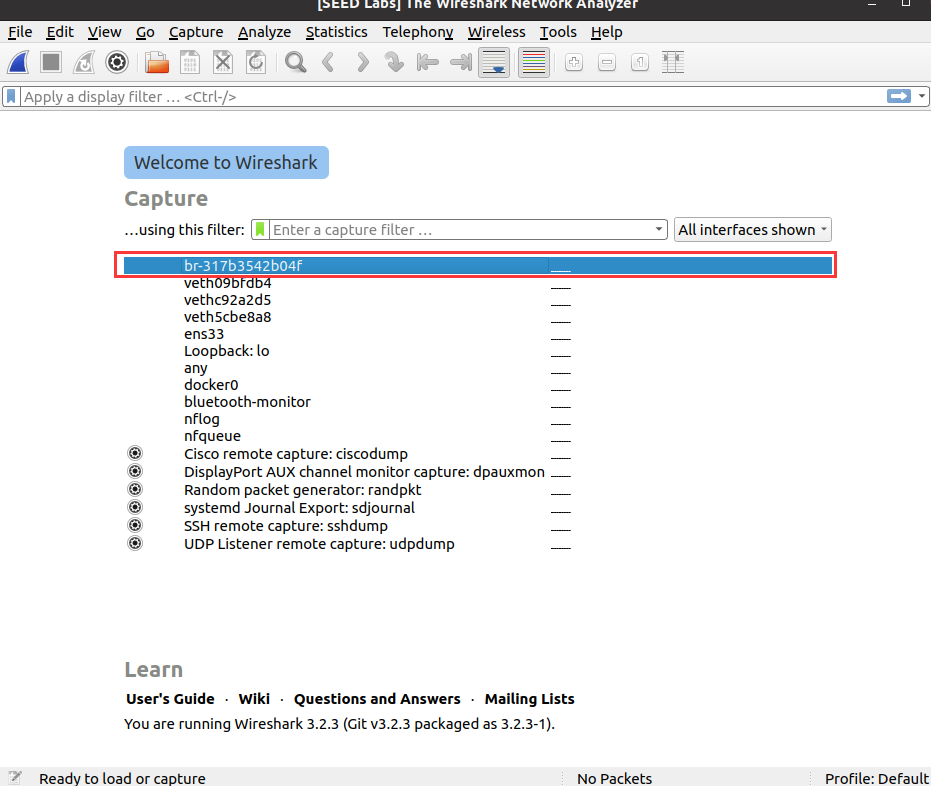
1. 使用其他主机（如ip为10.9.0.6）来telnet此victim主机（10.9.0.5），发现依然可以telnet通。说明，SYN Cookie机制成功抵御了泛洪攻击。



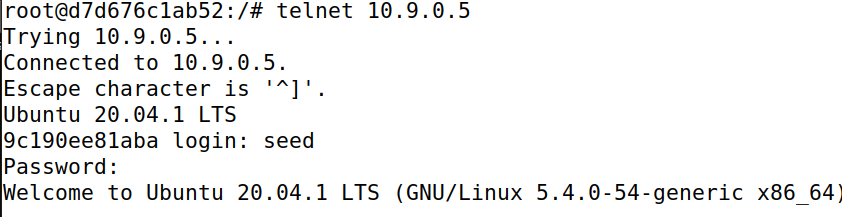
**Task 2: TCP RST Attacks on telnet Connections**

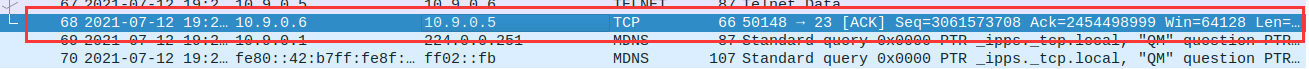
**——断开现有的telnet连接**

在wireshark中选择攻击机id。

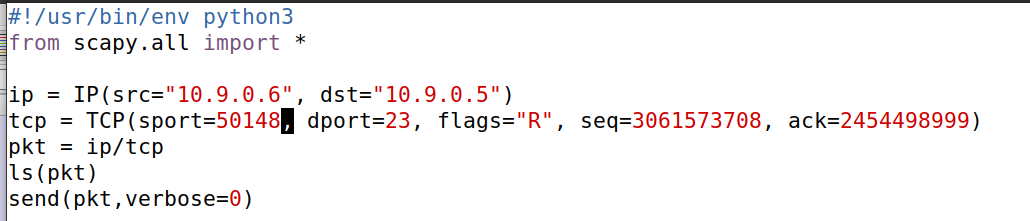


在主机B（10.9.0.6）上telnet此victim主机（10.9.0.5），



Wireshark中抓包得到：

在攻击机上新建ATTACKER.py文件：



**Task 3: Traceroute**

**Task4** **: Sniffing and-then Spoofing**