**实验四 TCP/IP Attack Lab**

**The Mitnick Attack Lab**

**实验报告**

57117202 朱惟璇

Lab1: Task 1，2，4

在本实验中，需要对TCP/IP协议进行攻击。可以在攻击中使用Netwox工具和/或其他工具。所有攻击都在Linux操作系统上进行。

为了简化TCP序列号和源端口号的“猜测”，假设攻击者与受害者位于同一物理网络上。因此，可以使用嗅探器工具获取该信息。以下是需要实施的攻击列表。

**1. Task 1: SYN Flooding Attack**

SYN泛洪是DoS攻击的一种形式，攻击者将许多SYN请求发送到受害者的TCP端口，但攻击者无意完成三向握手过程。攻击者使用欺骗性IP地址或不继续该过程。通过这种攻击，攻击者可以淹没受害者的队列，该队列用于半开的连接，即已完成SYN，SYN-ACK但尚未获得最终ACK的连接。当此队列已满时，受害者无法再建立任何连接。

队列的大小具有系统范围的设置。在Linux中，可以使用以下命令检查设置：

$ sudo sysctl -q net.ipv4.tcp\_max\_syn\_backlog

可以使用命令“ netstat -na”来检查队列的使用情况，即与监听端口关联的半开连接的数量。此类连接的状态为SYN-RECV。如果三向握手完成，则连接状态将建立。

在此任务中，需要演示SYN泛洪攻击。可以使用Netwox工具进行攻击，然后使用嗅探器工具捕获攻击数据包。进行攻击时，请在受害计算机上运行“netstat -na”命令，并将结果与攻击之前的结果进行比较。还请描述如何知道攻击是否成功。

用于此任务的相应Netwox工具编号为76。这是此工具的简单帮助屏幕。也可以键入“netwox 76 --help”以获取帮助信息。

**SYN Cookie对策：**如果攻击似乎没有成功，则可以调查的一件事是是否打开了SYN Cookie机制。SYN cookie是一种抵御SYN泛洪攻击的防御机制。如果计算机检测到受到SYN泛洪攻击，则该机制将启动。可以使用sysctl命令打开/关闭SYN cookie机制：

$ sudo sysctl -a | grep cookie (Display the SYN cookie flag)

$ sudo sysctl -w net.ipv4.tcp\_syncookies=0 (turn off SYN cookie)

$ sudo sysctl -w net.ipv4.tcp\_syncookies=1 (turn on SYN cookie)

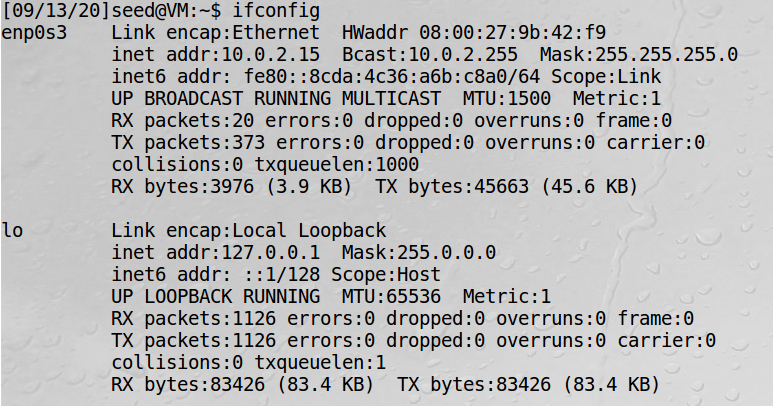
请在打开和关闭SYN cookie机制的情况下运行攻击，然后比较结果。在报告中请说明为什么SYN cookie可以有效地保护计算机免受SYN泛洪攻击。

**关于Scapy的注意事项：**尽管从理论上讲，可以使用Scapy来完成此任务，但已经观察到Scapy每秒发送的数据包数量比Netwox少得多。如此低的比率使攻击难以成功，无法使用Scapy成功进行SYN泛洪攻击。

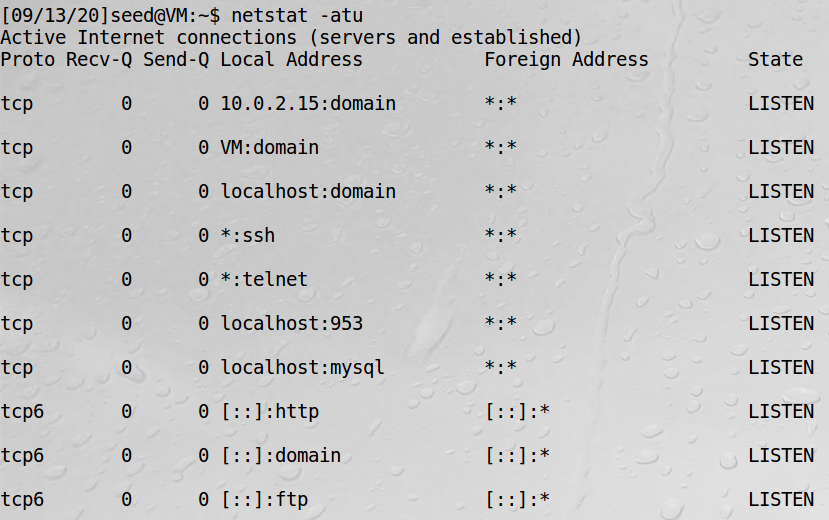
检查队列的大小：

C:\Users\zhuweixuan\AppData\Roaming\Tencent\Users\467718102\QQ\WinTemp\RichOle\H1ZDP664FR@N6{46}5QF]XG.png

设置三个虚拟机，如下图所示，虚拟机的ip地址可以从图上得出：

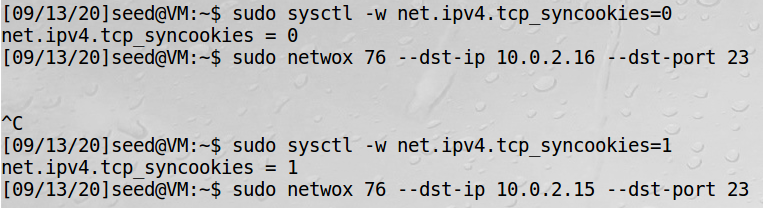


一般情况下ip地址为10.0.2.15主机的telnet连接如图：





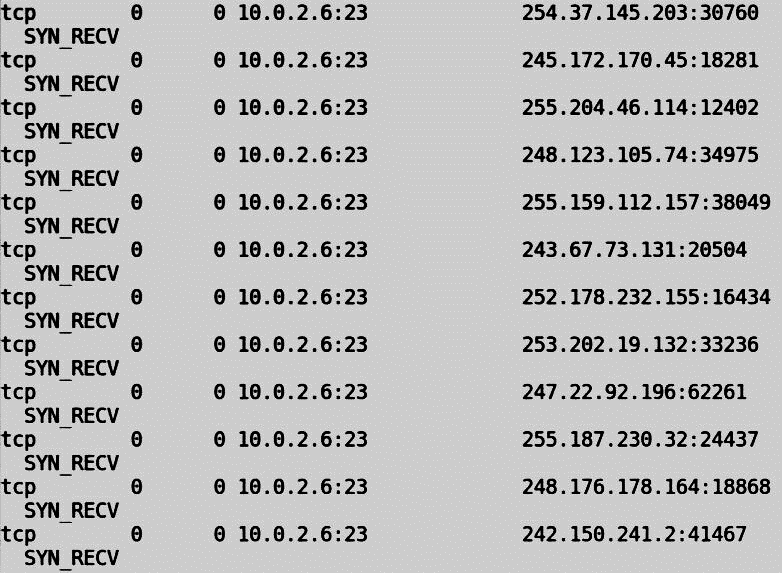
使用命令sysctl关闭SYN cookie机制:



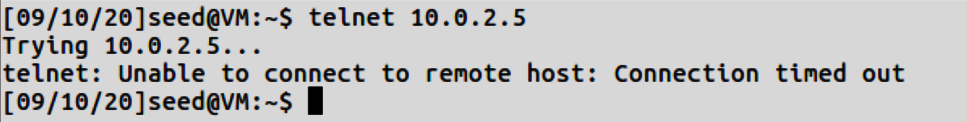
使用命令sudo netwox 76 --dst-ip 10.0.2.15 --dst-port 23进行SYN洪泛攻击：

C:\Users\zhuweixuan\AppData\Roaming\Tencent\Users\467718102\QQ\WinTemp\RichOle\YP3TT3{BOR12(O[K$$$YU}K.png

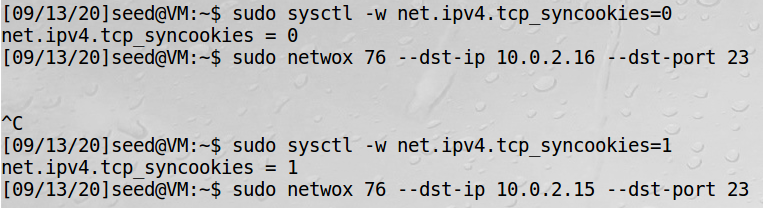
连接的状态发生改变：



再次尝试建立telnet连接，发现此时无法建立连接，说明SYN泛洪攻击成功：



再使用sysctl命令打开SYN cookie机制：



发现可以进行talent连接，说明开启后泛洪攻击失败。

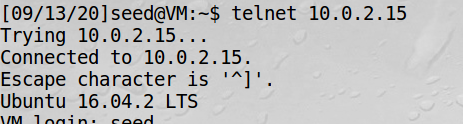
SYN Cookie是对TCP服务器端的三次握手协议作一些修改，专门用来防范SYN Flood攻击的一种手段。它的原理是，在TCP服务器收到TCP SYN包并返回TCP SYN+ACK包时，不分配一个专门的数据区，而是根据这个SYN包计算出一个cookie值。在收到TCP ACK包时，TCP服务器在根据那个cookie值检查这个TCP ACK包的合法性。如果合法，再分配专门的数据区进行处理未来的TCP连接，如果不合法，则SYN cookie可以有效防止SYN泛洪攻击。

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

TCP RST攻击可以终止两个受害者之间已建立的TCP连接。例如，如果两个用户A和B之间建立了telnet连接（TCP），则攻击者可以从A到B欺骗RST数据包，从而破坏此现有连接。为了成功进行此攻击，攻击者需要正确构造TCP RST数据包。

在此任务中，需要发起TCP RST攻击以断开A和B之间的现有telnet连接。此后，对ssh连接尝试相同的攻击。请描述观察。为了简化实验，假设攻击者和受害者位于同一个LAN上，即攻击者可以观察到A和B之间的TCP流量。

首先尝试进行telnet连接，可以连接成功：



使用sudo命令查看端口号为23

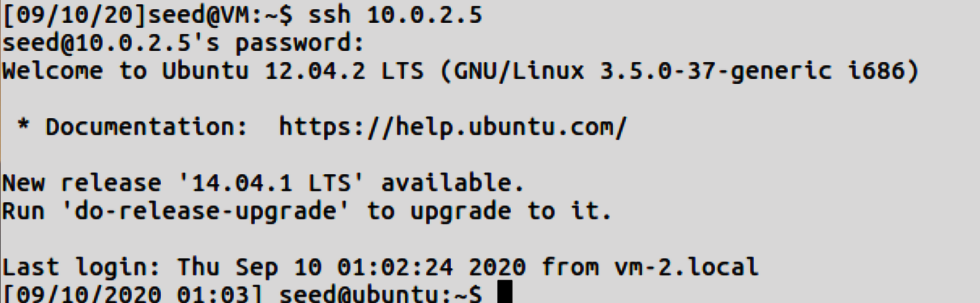
攻击者向受害者发送RST报文，企图中断观察者和受害者间的telnet连接。C:\Users\zhuweixuan\AppData\Roaming\Tencent\Users\467718102\QQ\WinTemp\RichOle\]VDMT0A{B5A74_H1%}YH(30.png

如图所示，执行攻击指令后，telnet连接终止，说明TCP RST攻击成功。



对ssh进行攻击：

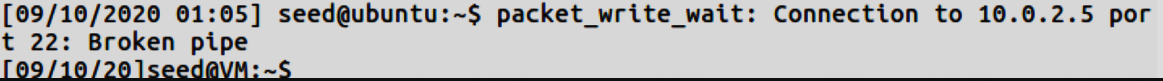
观察者先与受害者建立ssh连接：



发送RST报文，试图中断观察者和受害者间的ssh连接：

C:\Users\zhuweixuan\AppData\Roaming\Tencent\Users\467718102\QQ\WinTemp\RichOle\U~DA~7T]XMHL%L8D%@J4(WU.png

可以看到建立的ssh连接被终止TCP RST攻击成功：



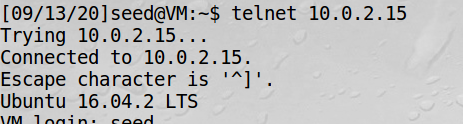
**4. Task 4: TCP Session Hijacking**

TCP会话劫持攻击的目标是通过向该会话中注入恶意内容来劫持两个受害者之间的现有TCP连接（会话）。如果此连接是telnet会话，则攻击者可以向该会话中注入恶意命令（例如删除重要文件），从而使受害者执行恶意命令。在此任务中需要演示如何劫持两台计算机之间的telnet会话。目标是使telnet服务器运行恶意命令。为了简化任务，假设攻击者和受害者位于同一局域网中。

可以使用Wireshark找出应该在欺骗的TCP数据包的每个字段中输入什么值。应该在SEED书的TCP会话劫持部分中指出，此处列出的命令并未设置TCP和IP标头的所有字段。在上面的netwox命令中，tcp-data部分仅获取十六进制数据。

首先在攻击者主机VM A上开启wireshark，设置过滤器，攻击者只抓受害者与观察者之间通信的报文。

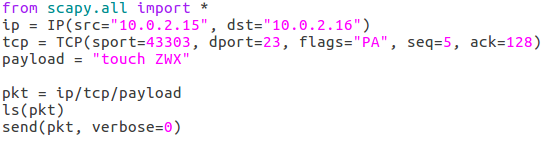
telnet远程连接10.0.2.15的虚拟机：



观察最后一个数据包（从观察者发往受害者）的源ip、目标ip、源端口、目的端口、sequence number和ack number，填入netwox伪造报文的对应位置。

可以看到端口为43303，最后一次通信，可以看到seq值为128，下一个seq为5

使用scapy进行攻击，新建hijack.py文件（netwox命令需要转成16进制，但是scapy攻击不用）：



可以在10.0.2.15的目录下面找到ZWX，说明已经劫持两台计算机之间的telnet会话，攻击成功。

