ARP MITM实践报告

姓名：任宇 学号：33920212204567

1. 实践题目

假设攻击者为M，攻击目标主机为A，A正在与B进行Telnet连接（A为Telnet Client、B为Telnet Server），M、A、B同在一个局域网，请在M上实现中间人攻击（MITM），截取记录A的操作。

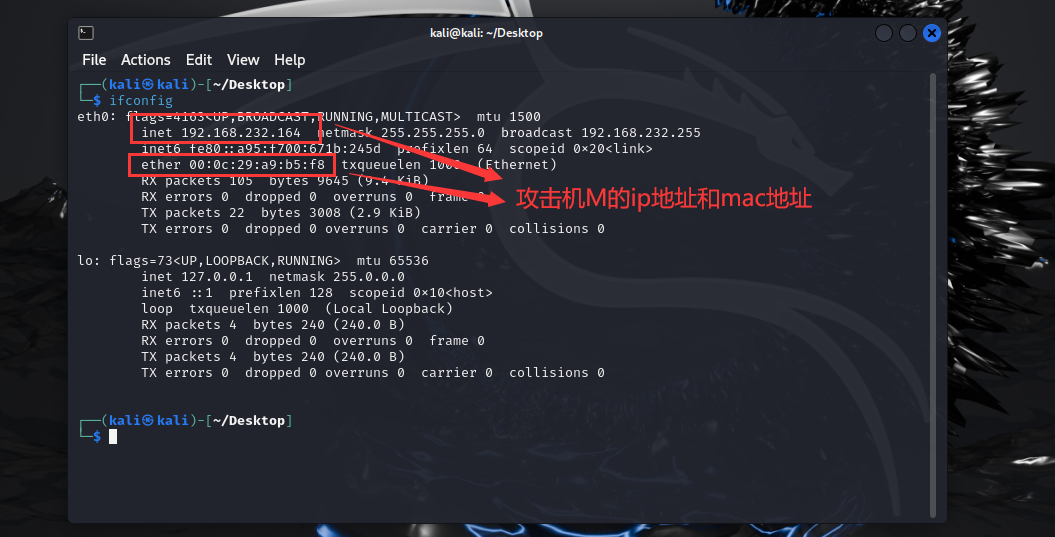
要求：

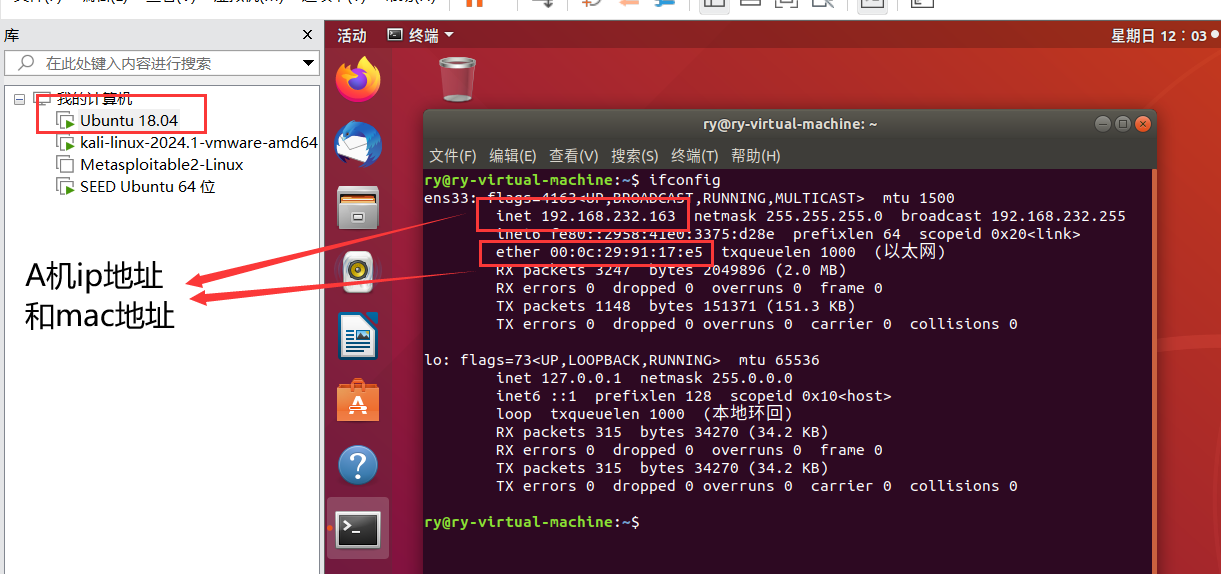
* 使用Scapy，编写成python代码实现；
* 观察并记录下各操作写入实验报告中；
* 成功截取记录A在Telnet中键入的操作。

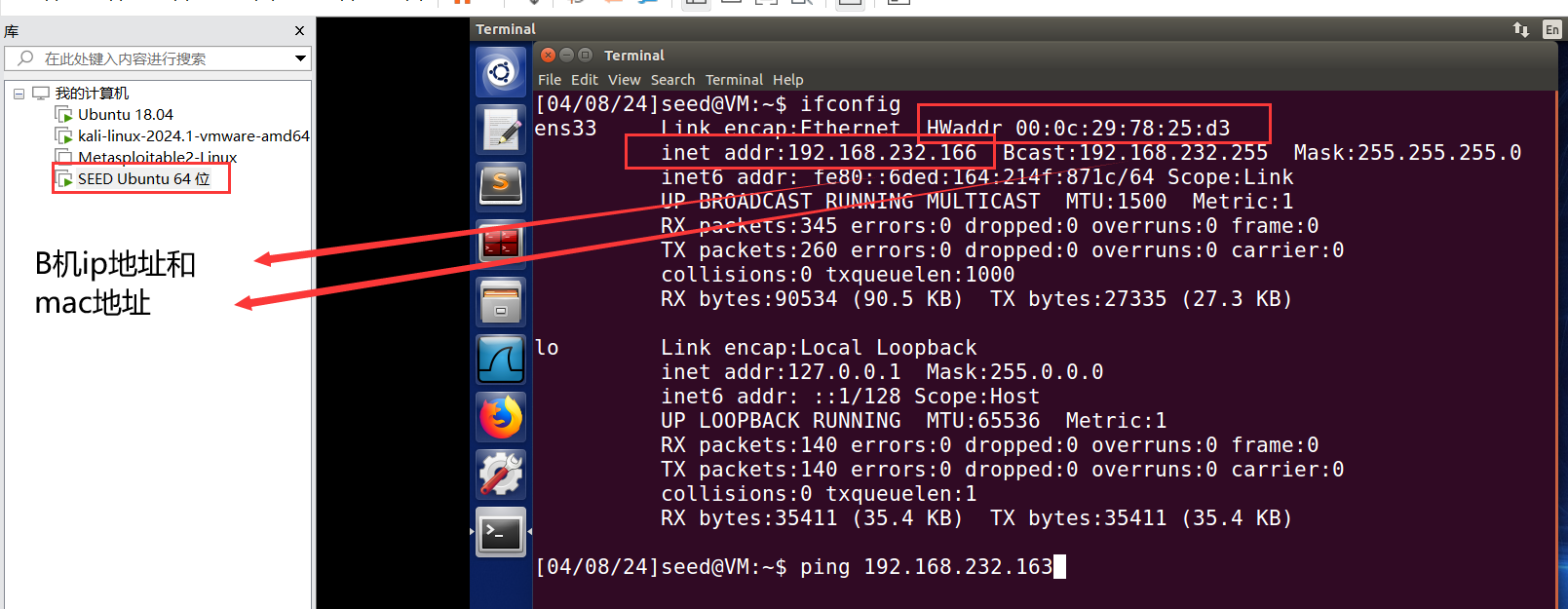
1. 实践环境

此次实践在虚拟机环境下进行，M机为Kali系统、A机系统版本为Ubuntu18.04以及B机系统版本为SEED Ubuntu 16.04。

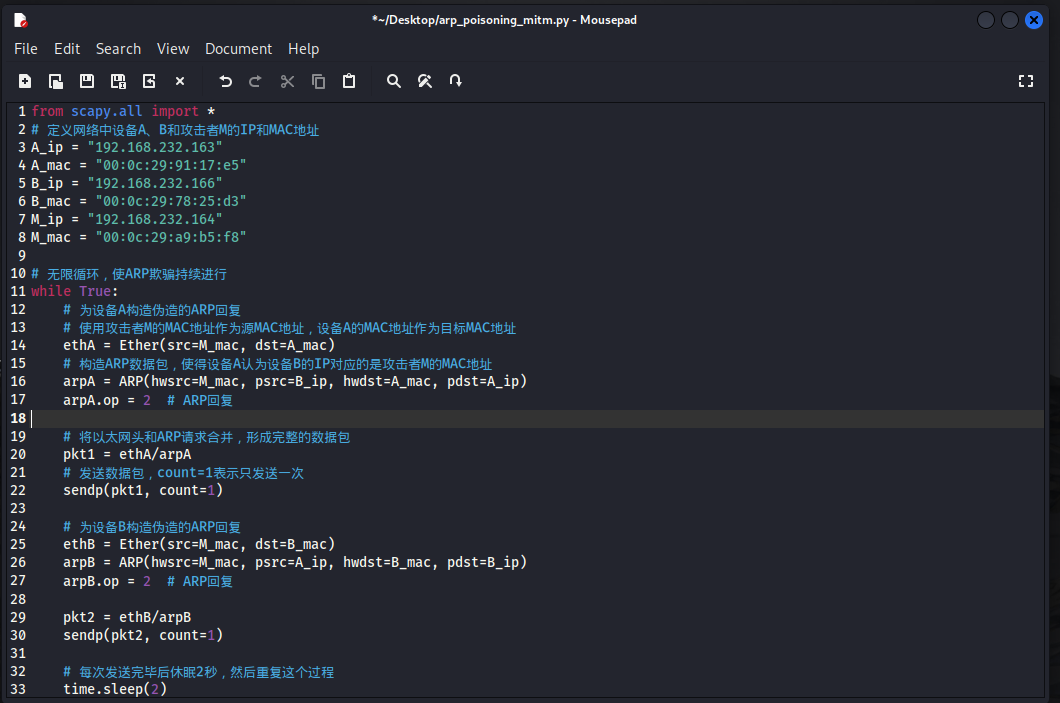
1. 实践步骤
2. 为了实现中间人攻击，首先需要启动ARP缓存攻击。首先，主机M需要对A和B进行了ARP缓存中毒攻击，这会使得在A的ARP缓存中，B的IP地址映射到M的MAC地址，在B的ARP缓存中，A的IP地址也将其映射到M的Mac地址。 在此步骤之后，在A和B之间发送的数据包将全部发送到M。实验过程中需要不断发送欺骗数据包（本实验为每2秒），否则假条目可能被真实的条目代替。
3. 获取M机、A机以及B机的ip地址和mac地址，以供编程：



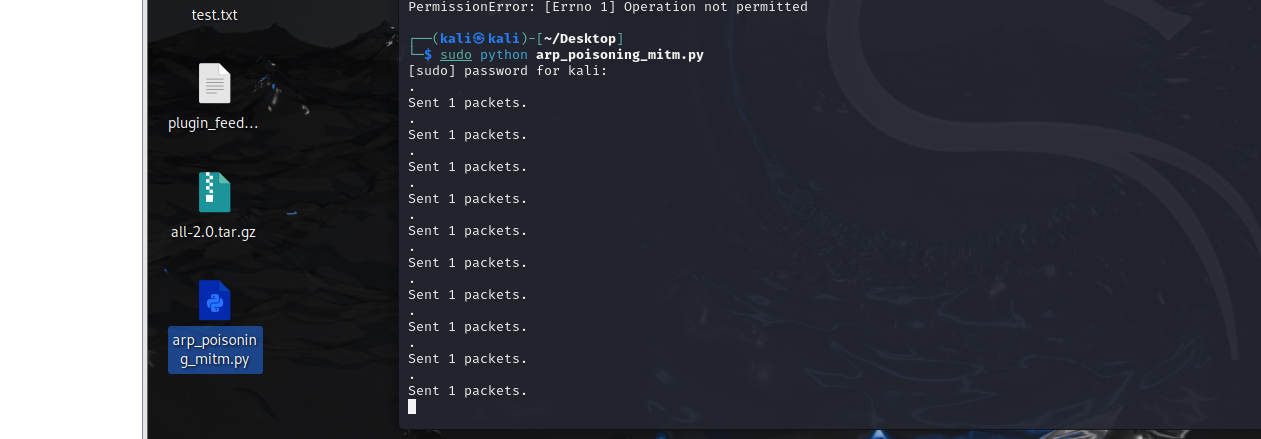


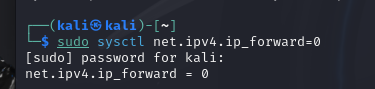


1. 使用Scapy库编写代码实现ARP缓存攻击：



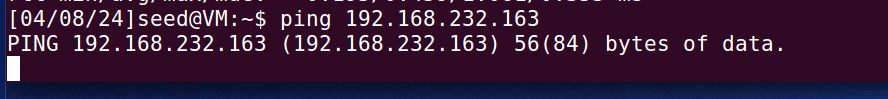
1. 启动程序，并将M机的ip转发关闭：



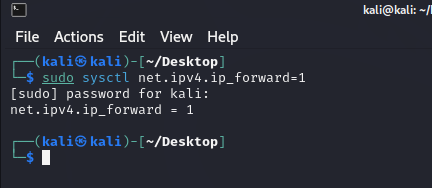


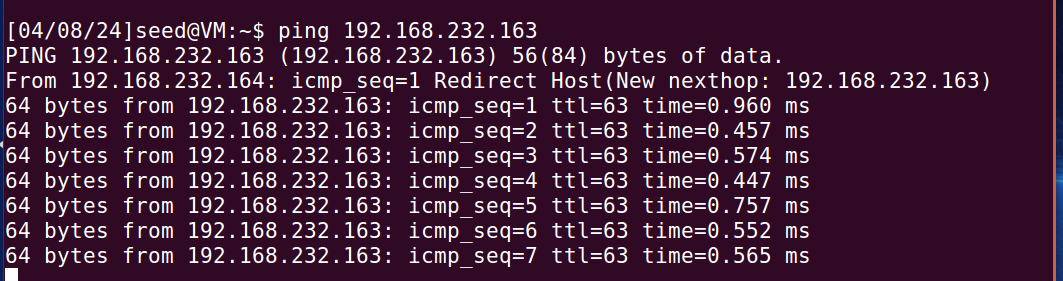
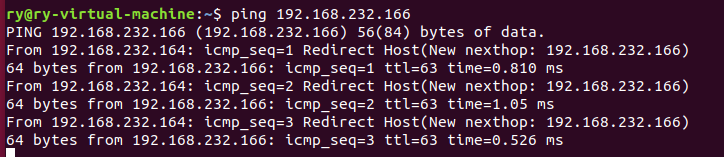
1. 此时A机和B机互相ping不通：



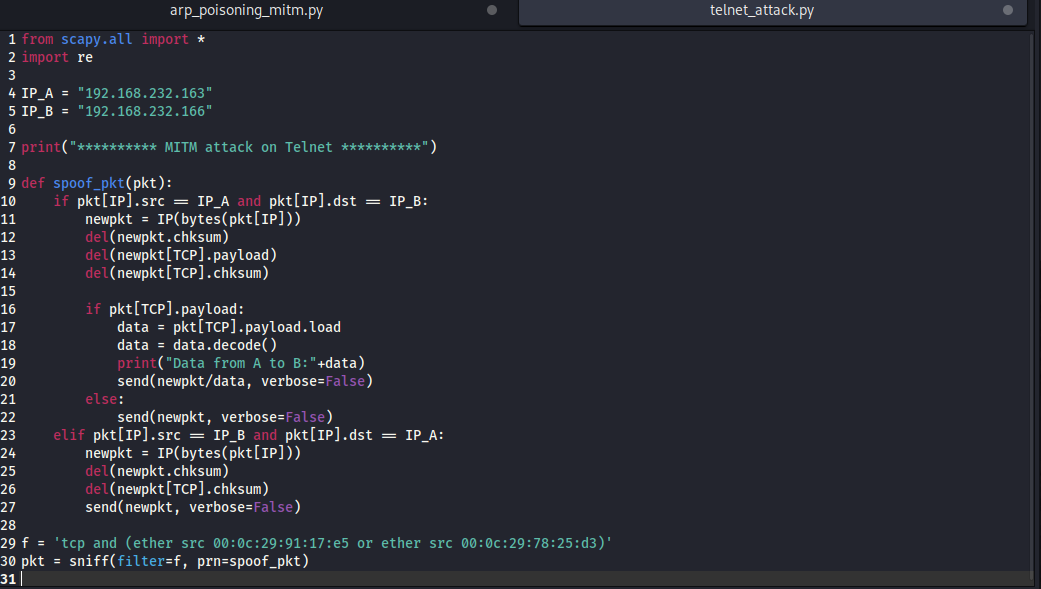


1. 打开M机ip转发，再尝试A机和B机互ping，可见此时ping通：

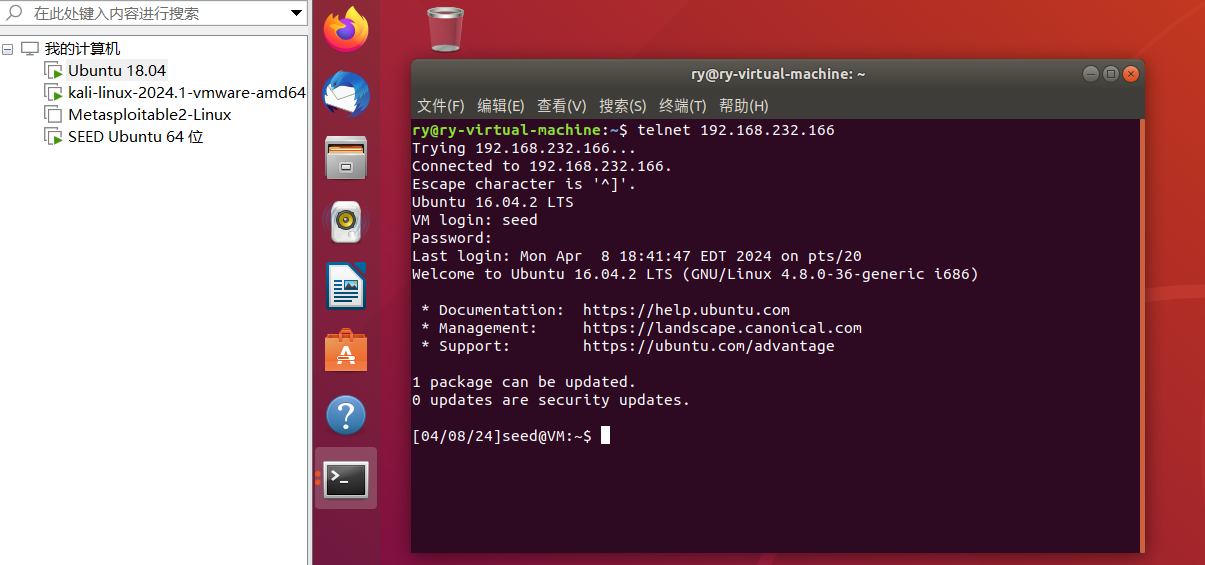




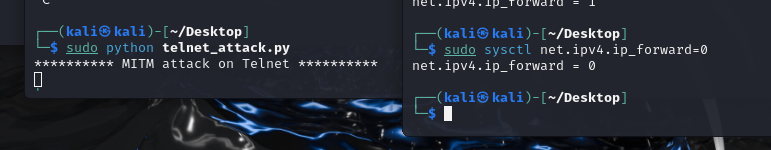
1. 使用Scapy库编写嗅探程序，此程序会截获从A机到B机的TCP数据包，打印出数据包的载荷（即传输的数据），然后将这些原始数据包重新发送。

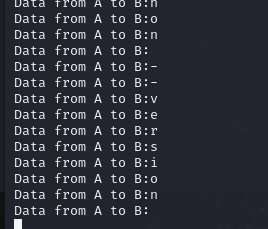
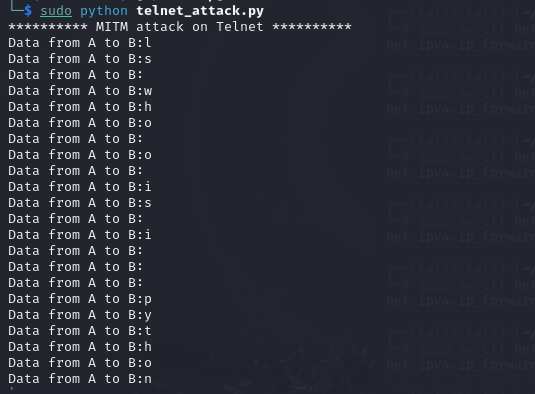
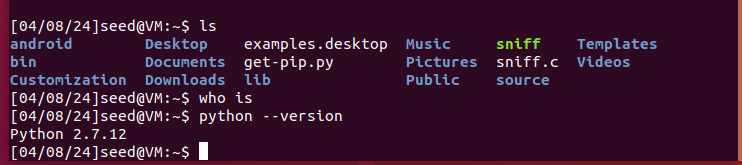


1. 首先保持M机的ip转发开启，因此可以在A机和B机之间成功创建telnet连接。



1. 成功创建telnet连接后，关闭ip转发，接着在M机上运行嗅探程序，此时，在A机上键入，可以发现程序成功截取A的键入操作。





* 关于telnet的补充说明：通常情况下，我们在telnet窗口中键入的每个字符都会触发一个单独的TCP数据包，但如果键入得非常快，一些字符可能会在同一个数据包中一起发送。这就是为什么在从客户端到服务器的典型telnet数据包中有效负载只包含一个字符。发送到服务器的字符将被服务器回显，然后客户端将在其窗口中显示该字符。因此，无论在客户端窗口中键入什么，在显示之前都要经过一个往返。

1. 实践总结

在这次的ARP中间人攻击实践中，我成功实现了对局域网内Telnet通信的监听和记录，特别是针对主机A作为Telnet客户端向B服务器进行的操作。通过使用Scapy库编写ARP缓存攻击和数据包嗅探脚本，我能够使所有从A到B的数据包首先通过攻击者机器M，从而实现了对通信的中间人劫持。

通过这个实践，我深刻理解了ARP缓存中毒攻击的原理及其在网络安全中的潜在危害。我了解到，基于文本的简单Telnet通信，通过中间人攻击能被轻易截获和记录，这揭示了在数据传输过程中加密的重要性。此外，实践中遇到的各种挑战，如维持ARP缓存中的假条目、嗅探指定网络流量包等，都加深了我对网络攻击技术和网络防御措施的认识。