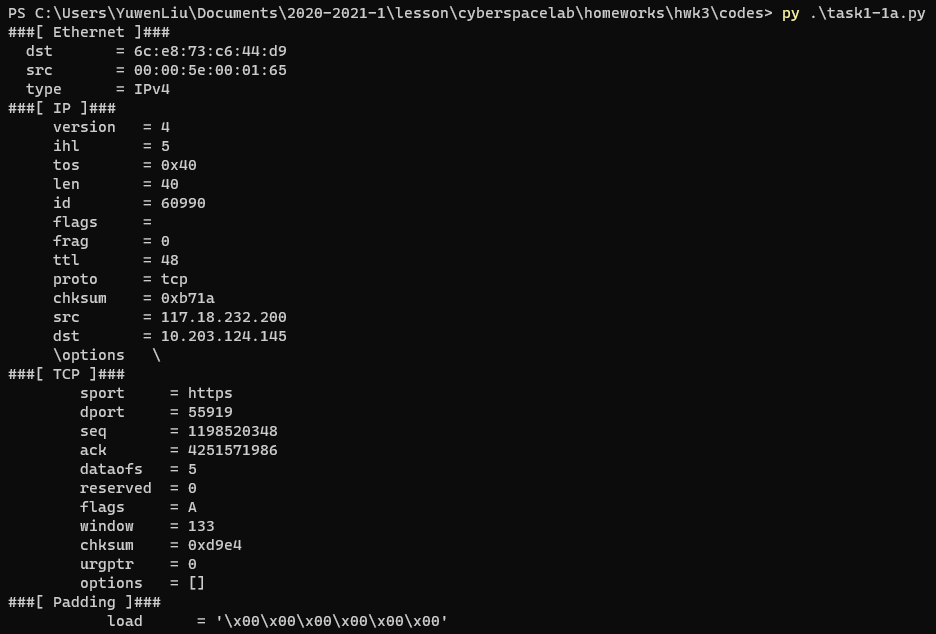
Packet Sniffing and Spoofing Lab

Task1-1

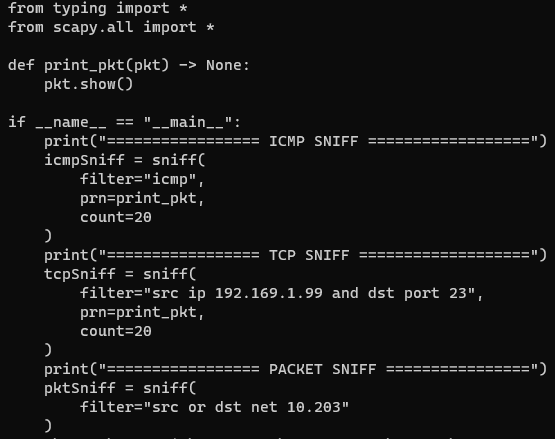
Task1-1a

直接运行代码就好了



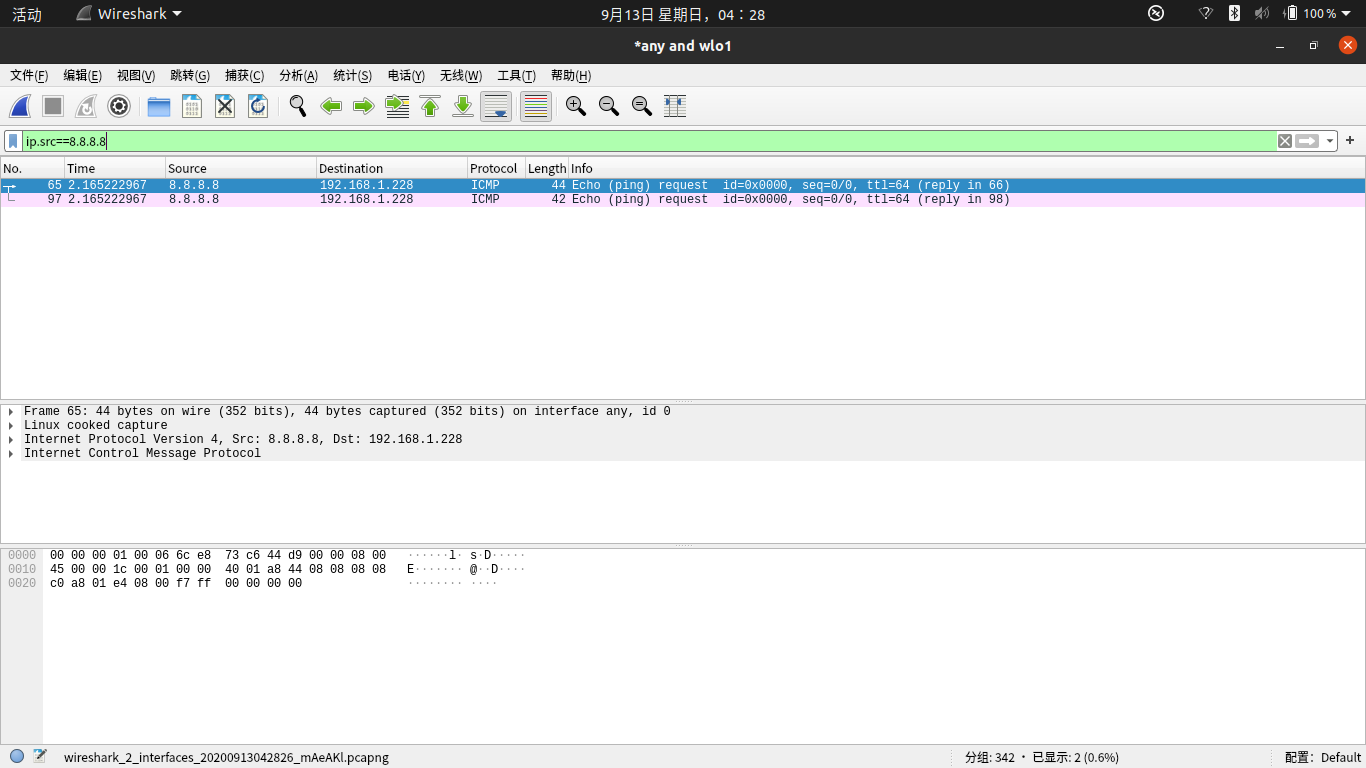
Task1-1b

对于不同的任务使用不同的过滤器



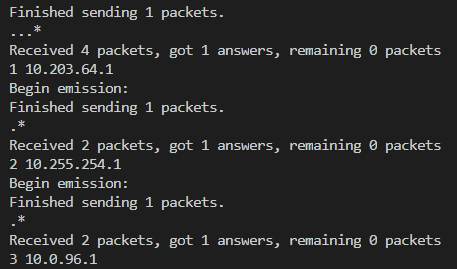
Task1-2 伪造ICMP报文\

伪造一个来自8.8.8.8的报文，显然不会从8.8.8.8向局域网发送ICMP的报文



Task1-3 Traceroute

使用ICMP报文，手动定义TTL值，每次加1，这样可以表示出报文在网络中的传输过程



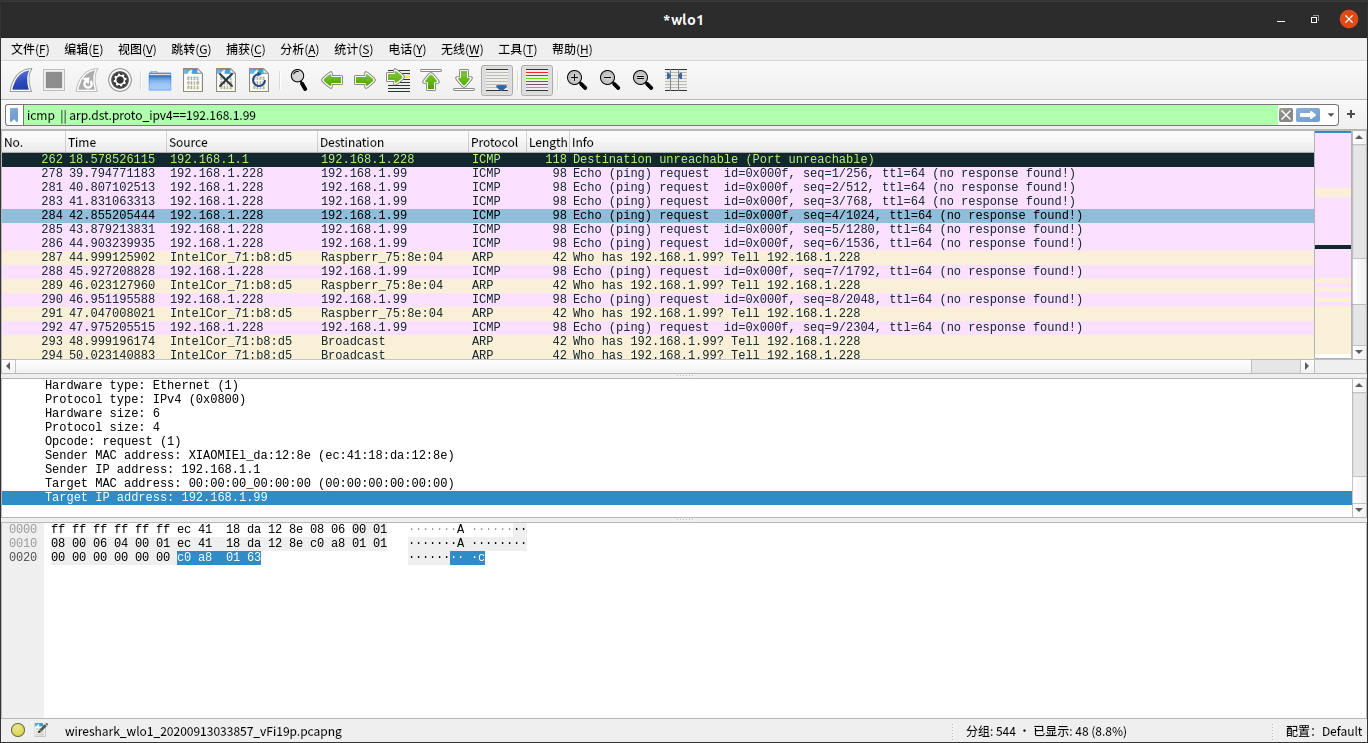
但是实际上这并不准确，因为每次都是发送一个新的报文，并不能确定这次的报文的路由和上一次报文的路由相同，因此，这只能是一个参考。

Task1-4

假设主机A和B，A ping B，现在的任务是，嗅探A向B发送的ping报文，一旦检测到，则立即给A回复，使得A认为B一直在线，尽管有时候B不在线。

该任务中，我通过Wireshark抓包发现，即使在混杂模式下，两台机器处于同一个局域网中，ping的ICMP报文依然无法被嗅探到。因此可能无法假冒B向A发送回复。

此外，在ping离线主机的时候，抓包发现，当ARP缓存表中相关信息未超时的时候，A会发送ICMP报文，但是发送几次接收不到回复的时候，A就会交替发送ARP报文和ICMP报文，来确认B的MAC地址是否正确。交替几次还是没有回应的情况下，就停止发送ICMP报文而只发送ARP报文，询问地址。此时观察ARP缓存表，发现缓存的MAC地址已经标为不完整。



因此，想要通过伪造ICMP伪装成离线主机，需要抓准时机，否则就需要ARP欺骗；而在伪装成在线主机，则被ping主机会收到多出来的ICMP回复，也可以一定程度上猜出有主机在伪装。

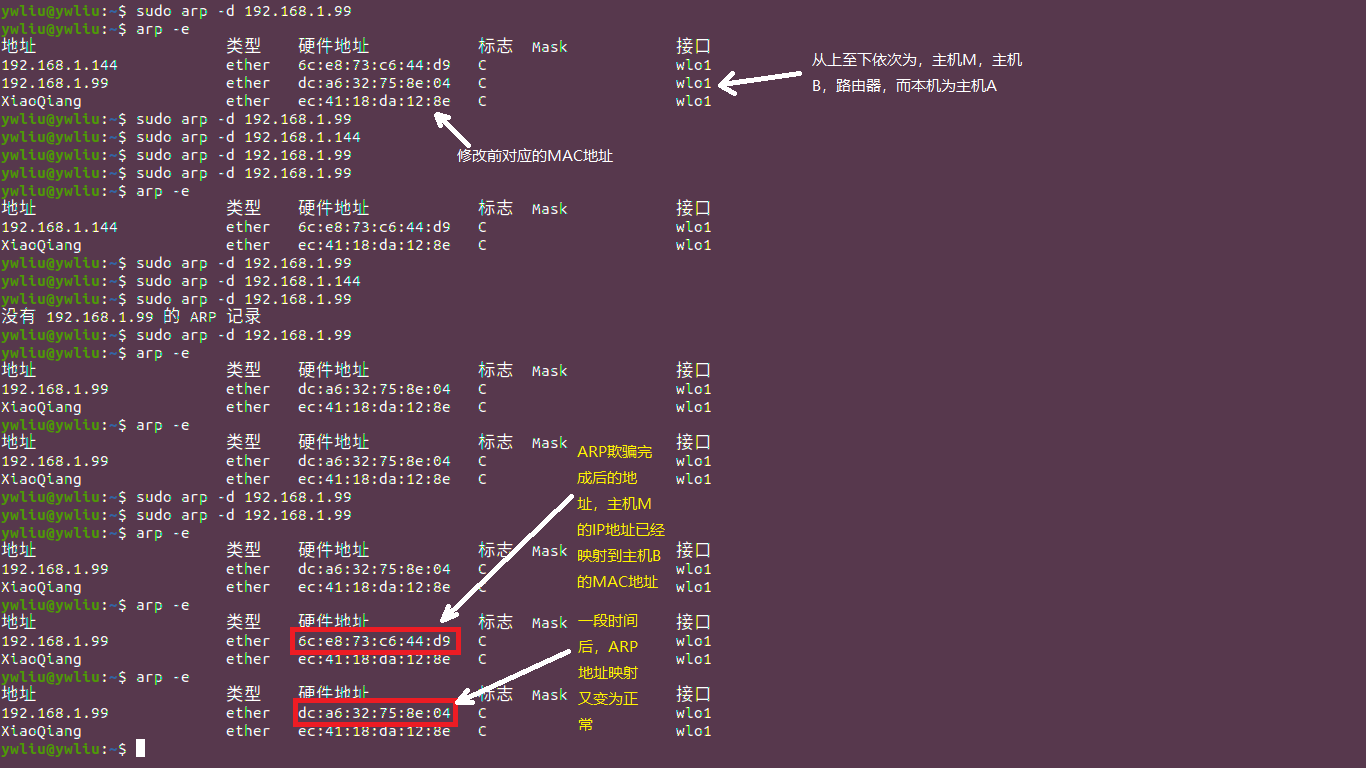
Lab2 ARP Cache Poisoning Attack Lab

Task1

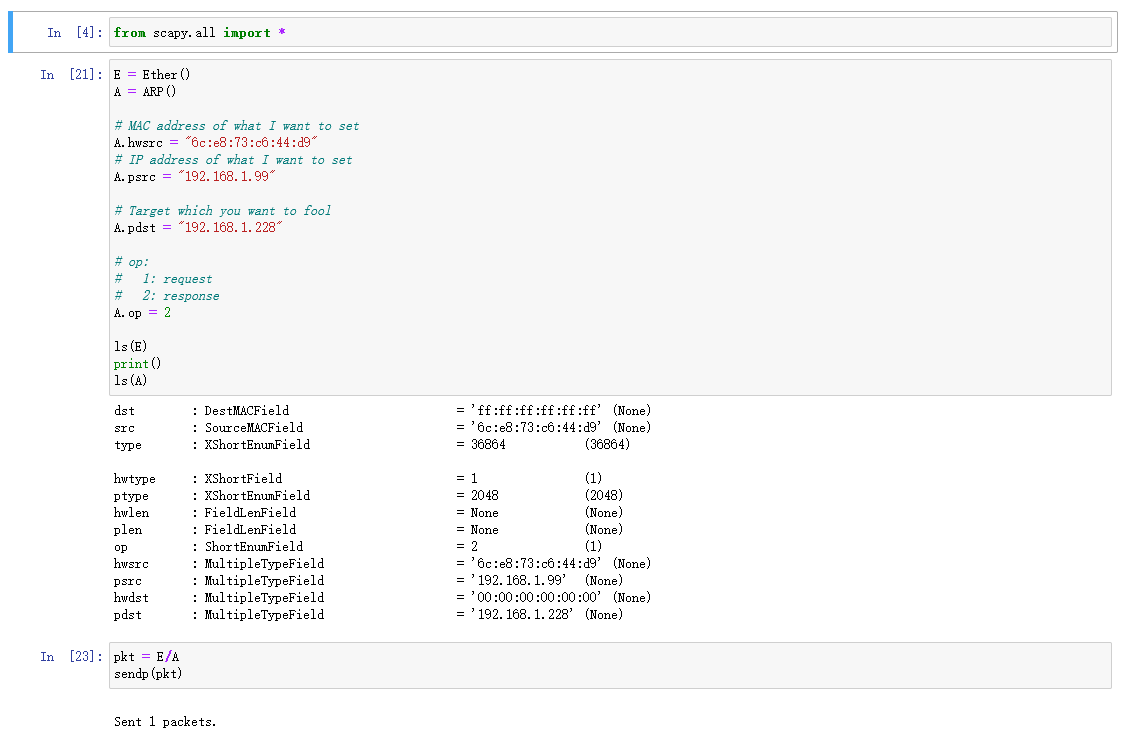
在该任务中，各个设备原始的MAC地址和IP地址如下

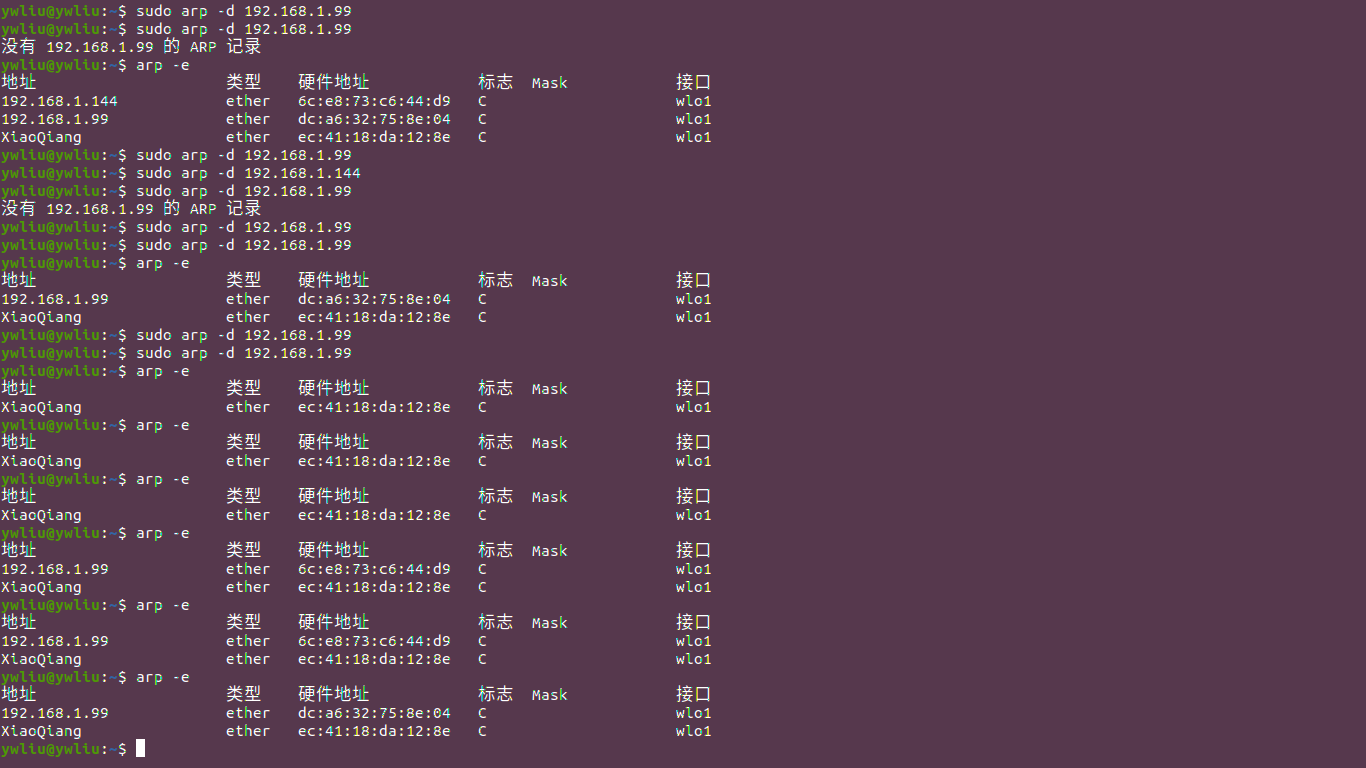
|  |  |  |
| --- | --- | --- |
| 设备 | MAC地址 | IP地址 |
| 主机A |  | 192.168.1.228 |
| 主机B | DC:A6:32:75:8E:04 | 192.168.1.99 |
| 主机M | 6C:E8:73:C6:44:D9 | 192.168.1.144 |

Task1A，使用ARP请求来实现ARP缓存攻击



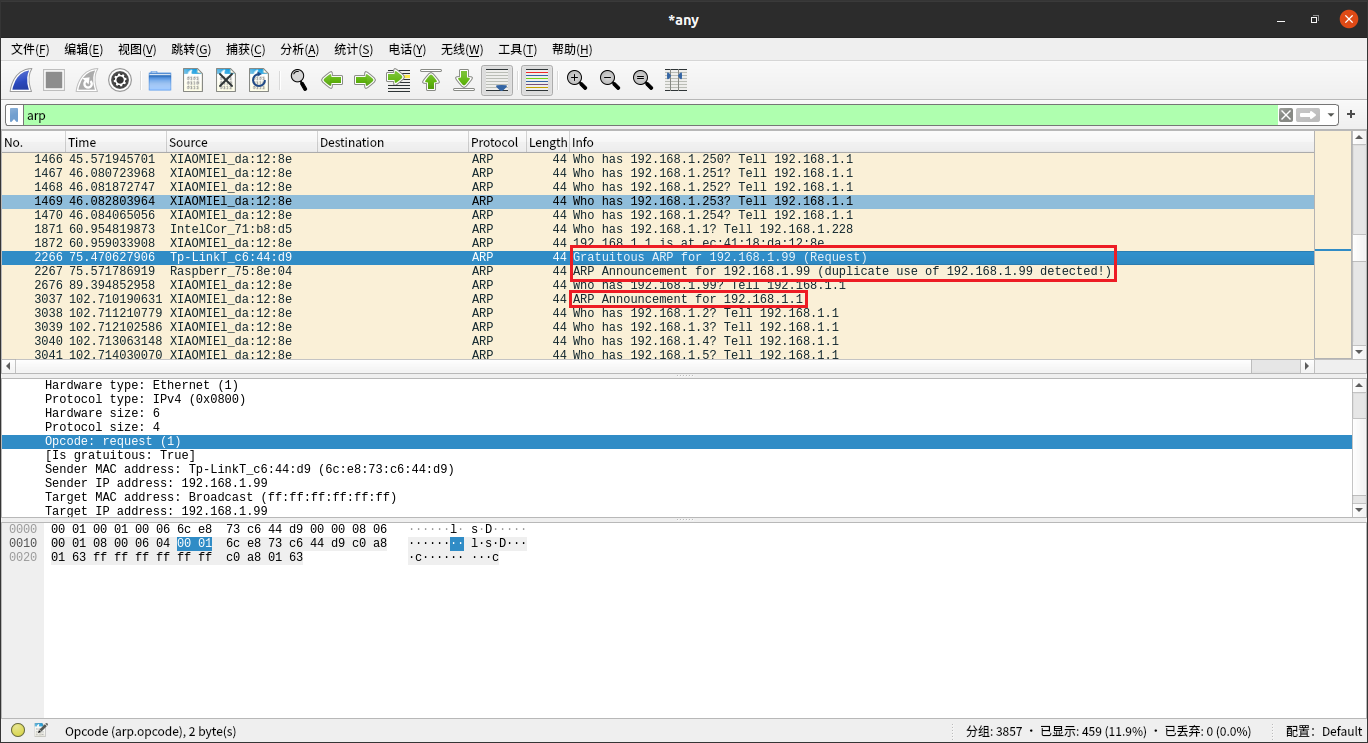
Task1B，使用ARP响应来实现ARP缓存攻击





和上图中标注的十分相似，也是同样完成了ARP缓存攻击

Task1C，使用ARP gratuitous信息



抓包观察发现，这里的确完成了攻击，在很短的时间内，通过gratuitous消息使得MAC地址得到了改变，但是很快被攻击的一方提出声明IP地址被映射到两个MAC地址上了，路由器开始ARP询问，然后广播询问结果，因此通过ARP gratuitous消息更改MAC地址的映射较难。

Lab3 IP/ICMP Attacks Lab

Task1 IP Fragmentation

Task1-a Conducting IP Fragmentation

按照要求，产生3个包，每个包负载32字节。算上首个包的头部8字节，共104字节。各个包的偏移量为：第一个包为0，第二个包为(32+8)/8=5，第三个包为5+32/8=9。此外除了最后一个包的flags为0，其余都为1。UDP报头只需要在第一个包中包含。



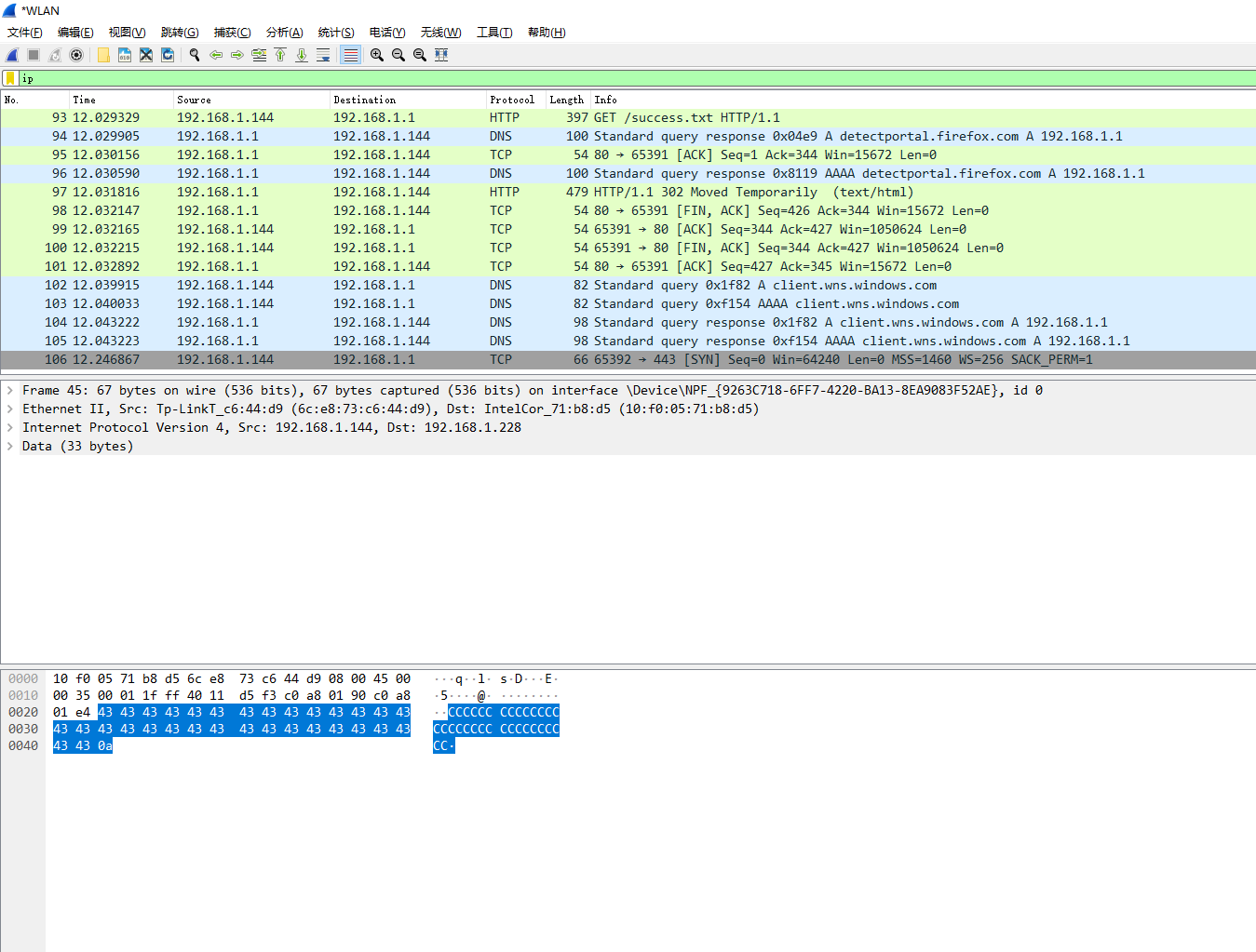
Task1-b

产生三个分段，使得其中会覆盖。



经过实验发现，后发送的分段会被先发送的分段覆盖，这一点有点不符合常规。

Task1-c



超长报文在发送的过程中就不能成功发出，图中抓包结果表明只有最后一个包被发送出去了

Task1-d

大量使用不完整的分段理论上会消耗服务器的大量资源，对服务器产生DoS攻击，但是在实际的操作过程中，并没有出现服务不可用的情况，原因可能是产生的分段不够多，或者操作系统将不完整的报文当成一个单独的IP报文，