IPID 哈希冲突查找实验报告

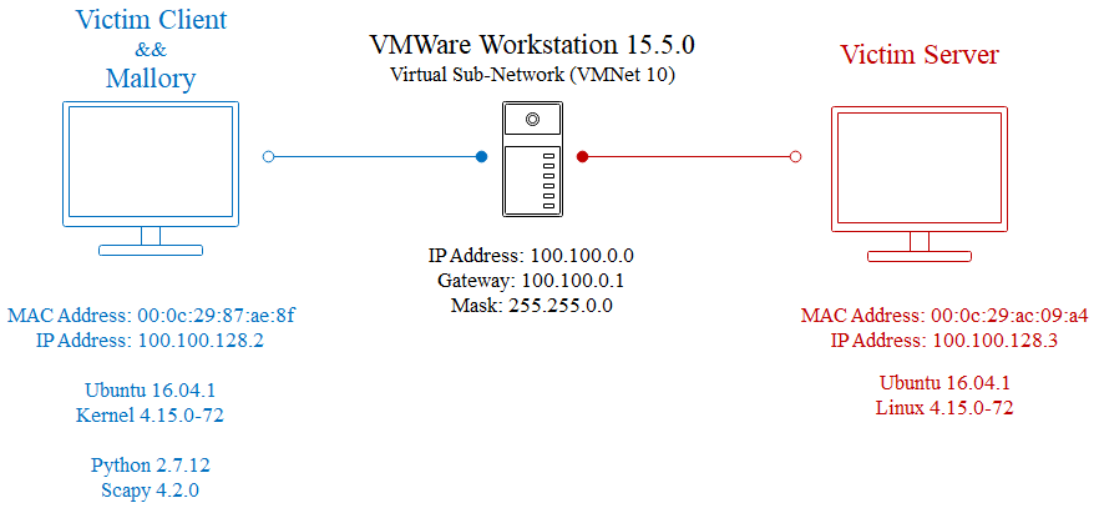
一． 实验目的：

本实验是 Off-Path TCP Exploit Via ICMP and IPID Side-Channel 的第一步，为攻击者 Mallory 确定 IP 地址以完成后续两步攻击。
本实验需要找到一个 IP 地址，使得：

Hash(Server_ip, Mallory_ip, ICMP, secert) = Hash(Server_ip, Client_ip, TCP, secert)

从而产生共享的 IPID 分配计数器，进而利用共享资源进行测信道攻击。
实验的步骤参考 PPT 当中的第 10 和 11 页。

二． 实验环境：



本实验进行在 VMWare 的虚拟组网下，使用了两台虚拟机。后续的实验程序全部运行在左侧的 Victim & Mallory 上。相关的实验参数见上图。

三． 实验步骤：

1. 实验文件：

single_check.py 用于检测某一个地址是否为哈希碰撞的地址。
parallel_check.py 用于在地址池当中查找出一个为哈希碰撞的地址。

2. 实验设计：

本部分详细讲述实验文件 parallel_check.py 的设计以及几个细节问题。

(1) 地址池的建立

地址池是候选的攻击者将要采用的地址，是可能产生碰撞的地址，是 Mallory 执行后续攻击可能采用的地址。

为方便实验进行，本文在同一子网 100.100.0.0 下选择若干地址作为地址池。但从幻灯片第 10 和 11 页可以看出，探查过程当中 Mallory 需要冒充候选地址发送并接收数据包。我们采用 **ARP 毒害**的方式使 Mallory 在该子网下具有冒充该候选地址进行收发报文的能力。

进行 ARP 毒害的方式分为 3 步见 parallel_check.py 中的 arp_inject 函数。

a. Mallory 以候选 IP 地址发送一个任意目的端口的 UDP 报文到 Server，由于 Server 上的对应端口并无服务，因而需要回复 ICMP 错误消息。
b. 由于候选 IP 主机实际上并不存在，Server 不知其物理地址，发送 ARP Request 消息进行 ARP 查询，Mallory 启动嗅探等待这个 ARP 查询消息。
c. 在嗅探到这个 ARP Request，Mallory 将自己的物理地址填入 ARP Reply 并发送。

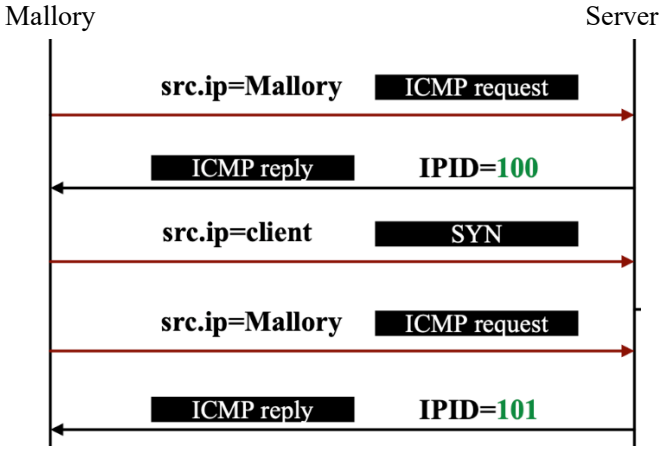
引入 ARP 毒害的方式仅仅是为了方便实验。

(2) 并发查找冲突

由于地址池当中地址数量较大，且报文交互耗时，必须实现**查找程序的并行化**。我们使用 Python 自带的多线程模块。每个线程的处理步骤如下：（check_new 函数）

a. 从地址池获取一个没被探查过的地址(line 111-129)
b. 对该地址进行 ARP 毒害，拥有伪装为该地址的能力(line 133)
c. 冒充 Victim 发送 ICMP 消息，强制 TCP 分片。方法见<Poisoning TCP-Based Applications via IP nFragmentation> IV 部分 B.(3) 与 Fig. 4。(函数 line 135-141)
d. 概率方式探查候选地址是否为冲突地址。调用 check_collision 来实现(line 143-153)

(3) 概率方式探查冲突



从幻灯片 10、11 页的流程上看出，攻击者伪造 IP 地址发送了两次 ICMP 期间夹杂了一次 TCP 消息。由于 IPID 计数器采用基于毫秒时间的均匀分布随机自增方式，若两次使用计数器时间间隔太长(超过 1ms)则可能产生超过 1 的自增，给实验带来误差。

在实践中发现夹在两次 ICMP 中间的 TCP 消息的发送可能带来上述的实验误差(存在但概率不超过 50%)，这时两次 IPID 的差值>=2，但事实上并未产生冲突。

我们的解决方案是**概率性冲突探查**。

动机是：实验中两次 IPID 差>=2 的不一定是哈希碰撞造成的，有可能是时间差造成的大于 1 的自增。但是如果产生了两次 IPID 差==1 的现象，则一定不是冲突，可以排除该地址。
具体做法是：将上图的报文交互进行多次，如果发现存在 IPID 差为 1 的则判定一定不是冲突，全部大于 1 认为其是 P 概率-冲突。再对该地址重复 N 次检测，全部检测通过则有足够大的把握认为其是冲突。

3. **实验流程：**
   调节 parallel_check.py 文件当中与地址池范围相关的参数，和并行线程数等参数。运行一段时间后将输出冲突地址。

   将冲突地址输入 single_check.py 当中可以手动验证输出的地址为冲突地址。

4. **实验效果：**
   在 20 线程并发的条件下:
   检测每个地址池中的地址用时约 0.076 秒。
   每秒大约检测 13 个候选地址。
   运行 3 分钟即有 40%左右的概率得到冲突地址。

四． 结论：

1. 通过目前的实验现象客观判断，幻灯片当中的找寻冲突地址方案是可行的。
2. 事实上，找到冲突地址的概率并没有想象中的那么小。
3. 将攻击在广域网上实施可能遇到的问题:（1）如何获得一个地址池（2）如何处理时延带来的计数器大于 1 自增的问题。
4. 效率上仍存在进一步优化的空间。