tcpreplay

将PCAP包重新发送，用于性能或者功能测试

补充说明

简单的说， **tcpreplay** 是一种pcap包的重放工具，它可以将用ethreal□wireshark工具抓下来的包原样或经过任意修改后重放回去。它允许你对报文做任意的修改（主要是指对2层、3层、4层报文头），指定重放报文的速度等，这样tcpreplay就可以用来复现抓包的情景以定位bug□以极快的速度重放从而实现压力测试。

选项

-A "<args>" 在使用 tcpdump 风格打印输出信息时，同时再调用tcpdump中的参数， 默认已经带有“-n,-l”□所以一般看到的都是ip地址，而没有主机名的打印，注意这个是在tcpreplay使用了-v参数时才能使用，不带-v不会报错，但是没有实际意义。格式□-vA “nnt”表示以tcpdump风格输出报文信息，并且不打印时间戳、主机名、端口服务名称。注意不要使用-c参数来指定打印的数据报文的个数，这样发送出去的报文也会变少。
-c <cachefile> 双网卡回放报文必选参数，后面紧跟cache文件名，该文件为tcpprep根据对应的pcap文件构造出来。
-D 把应用层的数据，使用dump mode写入到指定文件中去，和-w□-W 参数一起使用。
-e <ip1:ip2> 指定端点的ip□即把发送报文的和接收的报文的ip都修改称对应的参数值中指定的ip□但是这样发送的出的报文不会区分client和server□□
-f <configfile> 指定配置文件。
-F 在发送报文时，自动纠正错误的校验和。对测试DUT的校验和检验。
-h 显示帮助文件。
-i <nic> 双网卡回放报文必选参数，指定主接口。
-I <mac> 重写主网卡发送出报文的目的MAC地址。
-j <nic> 双网卡回放报文必选参数，指定从接口。
-J <mac> 重写从网卡发送出报文的目的MAC地址。
-k <mac> 重写主网卡发送报文的源MAC地址。
-K <mac> 重写从网卡发送报文的源MAC地址。
-l <loop> 指定循环的次数。
-L <limit> 指定最大的发包数量。可以在确认连接的调试时使用。
-m <multiple> 指定一个倍数值，就是必默认发送速率要快多少倍的速率发送报文。 加大发送的速率后，对于DUT可能意味着有更多的并发连接和连接数，特别是对于BT报文的重放， 因为连接的超时是固定的，如果速率增大的话， 留在session表中的连接数量增大，还可以通过修改连接的超时时间来达到该目的。
-M 表示不发送“ 火星 ”的ip报文□man文件中的定义是 0/8、172/8、 255/8。
-n 在使用-S参数，不对混杂模式进行侦听。
-N <CIDR1:CIDR2,...> 通过伪造的NAT□重写IP地址。这个参数应该有很重要的应用，目前没有测试使用。
-O 没有测试使用。
-p <packetrate> 指定每秒发送报文的个数，指定该参数，其它速率相关的参数被忽略，最后的打印信息不会有速率和每秒发送报文的统计。
-P 表示在输出信息中打印PID的信息，用于单用户或单帐户模式下暂停和重启程序。
-r <rate> 指定发送的速率。目前-m/-r/-p这3个参数的相互关系。
-R 让网卡极限速度发数据包。
-t <mtu> 指定MTU□标准的10/100M网卡的默认值是1500。
-T Truncate packets > 截去报文中MTU大于标准值的部分再发送出去，默认是不发送□skip掉。
-v 每发送一个报文都以 tcpdump 的风格打印出对应的信息。

```
-V 查看版本号。
-w <file> 将主网卡发送的报文写入一个文件中，参数后紧跟文件名。
```

实例

1、重放在客户端 ftp 连接的报文

a□在客户端使用 ethereal 抓包，存为 ftp.pcap 文件。

b□ 将 ftp.pcap 文件进行 tcpprep 操作，制作 cache 文件。

```
[root@A ~]# tcpprep -an client -i ftp.pcap -o ftp.cache –v
```

c□ 将 DUT 设备的两个接口和 PC 的两个接口使用网线连接，使用 tcpreplay 重 放报文。注意防火墙的配置为网桥（透明）模式。

```
[root@A ~]# tcpreplay -c ftp.cache -i eth0 -j eth1 ftp.pcap -R –v
```

-R 参数表示全速发送□-v 显示打印信息。

2、重放在客户端 BT 连接的报文

a□在实验室 BT 下载一些台湾的娱乐节目和热门的大片，使用 ethereal 抓包， 存为 bt.pcap 文件。注意 pcap 文件大小的控制，对 pc 的内存要求比较高，我保 存了一个 600 多 M 的 pcap 文件用了 40 多分钟，大家有需要可以直接从实验室 copy□

b□将 bt.pcap 文件进行 tcpprep 操作，制作 cache 文件。

```
 [root@A ~]# tcpprep -an client -i bt.pcap -o bt.cache -C "100M BT Packet"
–v
```

制作 cache 文件，在 cache 文件中写入“100M BT Packet”的注释。

c□使用 tcpreplay 重放报文。

```
[root@A ~]# tcpreplay -c bt.cache -i eth0 -j eth1 bt.pcap -v –R
```

3、重放 tftp 服务器上抓到的报文

a□在 tftp 服务器上使用 ethereal 抓包，存为 tftp.pcap 文件。

b□将 pcap 文件进行 tcpprep 的操作，制作 cache 文件。

```
[root@A ~]# tcpprep -an server -i tftp.pcap -o tftp.cache –v
```

注意：我在测试的时候犯了一个错误，使用 DUT 的 tftp 升级来做实验，同时穿 过 DUT 重放报文，结果在网卡发送报文的后□DUT 的 mac 地址做了的回应，导致 交互过程没有穿过 DUT□这个问题比较搞笑，上午弄了半天才发现原因，开始还 以为 udp 的连接不能重放。

c□使用 tcpreplay 重放报文。

```
[root@A ~]# tcpreplay -c tftp.cache -i eth0 -j eth1 tftp.pcap –v
```