流量拷贝工具tcpcopy测试部署——流程整理和问题小结

前期工作

1 步骤

1.1 知识储备

- tcp/ip协议【TCP/IP协议整理——报文/流程解析】
- websocket协议【Websocket协议整理】
- tcpcopy功能及架构【*readme of tcpcopy*】

1.2 环境搭建准备

1.2.1 主机

- target server（流量拷贝目标）：10.12.7.9
- online server：10.12.8.30
- assistant server：10.12.7.8

1.2.2 工具

- tcpcopy version1.2.0
- intercept version1.0.0

1.2.3 服务

- 实时短语音私有云服务

服务布置

1 流程步骤

1.1 拉服务包

1. 服务打包在8.13机器上，7.9和8.30都需要布置服务；
2. 使用scp 拉取/推送服务包；
3. 解包。

scp command

```
pullscp (-P <PORT>) <USR>@<IP>:<remote filepath> <local dirpath>
scp -P 5837 xionghao.li@10.12.8.13:/home/xionghao.li/tcpcopy.zip /home


pushscp <local filepath> (-P <PORT>) <USR>@<IP>:<remote dst dirpath>
scp ./tcpcopy.zip -P5837 xionghao.li@101.12.8.13:/home/xionghao.li

-r:
usrroot
```

1.2 运行服务

1. 修改docker-compose.yaml中volumes参数和创建对应路径，保证资源和日志挂载成功；
2. 修改docker-compose.yaml中casr参数拉取所需服务镜像；
3. 修改docker-compose.yaml中ports参数绑定主机端口；（7.9上服务端口为58002，8.30上服务端口为38002）
4. 在docker-compose.yaml的目录下运行docker-compose up -d启动服务容器；
5. docker ps|grep <keyword>检查服务是否启动。

1.3 测试服务连通性

用任意主机跑测试脚本，观察容器内的服务日志，看看是否连通识别服务。

2 问题小结

2.1 在运行服务的过程中，7.9上的服务起不来【阻塞时长：0.5 hours】

原因：7.9这个主机的网络连通性较差，无法连接到镜像仓库服务器上，镜像拉取失败导致服务启动失败。

解决方法：

1. 找到同时连通仓库和7.9的服务器的主机（8.13），docker pull所需要镜像；
2. docker save <image ID> -o <filename>.tar将镜像保存为tar包；
3. 通过scp指令将镜像tar包传至7.9上；
4. docker load -i <image tar>将镜像加载，注意镜像的名字是<none>，与yaml文件内镜像不匹配
5. docker tag <image ID> <name:tag> 修改成对应镜像名
6. 拉起服务

tcpcopy部署——在线流量回放

1 流程步骤

1.1 tcpcopy环境布置

tcpcopy布置到online server 10.12.8.30上

intercept布置到assistant server 10.12.7.8上

1.2 设置target server路由

set route

```
route add -net 10.12.7.0 netmask 255.255.255.0 gw 10.12.7.8
```

将target server(7.9)上所有发送到7.0网段上的包都路由到assistan server(7.8)上

1.3 配置/编译/启动intercept

1. assistant server需要关闭路由转发功能，echo 0 > /proc/sys/net/ipv4/ip_forward；
2. 在intercept路径下,顺序执行./configure <option> →make→ make clean，编译intercept
3. route指令查看路由表，找target server的ip的对应项的网卡；<pcap filter statment> 填tcp and src port 58002即可抓取target server(7.9:58002)的tcp响应包

run intercept

```
./usr/local/intercept/sbin/intercept -i <> -F '<pcap filter statment>' -d
```

1.4 配置/编译/启动tcpcopy

1. 在tcpcopy路径下，按序执行./configure <option> →make→ make clean，编译tcpcopy；
2. 指定网卡同1.3.3，<transfer>格式为 localServerPort-targetServerIP:targetServerPort，拷贝得到的包源IP修改为<fakedIP>，以此设置target server的路由；
3. tcpcopy运行必须在intercept之后，否则一定失败

run tcpcopy

```
./usr/local/intercept/sbin/tcpcopy (-i <>) -x <transfer>  -s <assistantServerIP> -c <fakedIP> -d
```

1.5 测试tcpcopy是否生效

1. 进入7.9和8.30的服务容器，追踪日志

2. 执行脚本向online server(8.30)申请识别服务
3. 如果online server 和target server都回应了识别服务请求说明tcpcopy工具生效

测试结果：

"recordId":"0561a525598711eca4e6e4b97a30e7cc"}, client: 10.12.7.141, server: , request: "GET /runtime/v1/recognize?res=aienglish-mix&productId=12345_casr HTTP/1.1", host: "10.12.8.30:38002"
2021/12/10 15:01:43 [info] 19#0: *2132 [lua] output.lua:0: result(): {"applicationId":"14796936228595de","source":"LITE","URL":"\/runtime\/v1\/recognize?res=aienglish-mix&productId=12345_casr","audioUrl":"http:\/\/records.unused.com\/0561a525598711eca4e6e4b97a30e7cc.wav","remoteIP":"10.12.7.141","time":"2021-12-10 15:01:43","params":{"app":{"applicationId":"14796936228595de","productId":"278583338"},"audio":{"sampleBytes":2,"audioType":"wav","sampleRate":16000,"channel":1},"request":{"skillList":[{"skillName":"utterances","skillId":"22372fce-4716-11ea-a43d-0f7f760adc83","version":"latest"}],"dictList":{},"coreType":"cn.asr.rec","customITNLexFile":"\/opt\/aispeech\/olive\/res\/asr_post_res\/test_itn\/custom_itn_lex.bin","buildinDictRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","res":"aitmp","recordId":"0561a525598711eca4e6e4b97a30e7cc","phraseFile":"\/opt\/aispeech\/olive\/res\/phrase_file_res\/48154546-8f69-11ea-a5e8-a7d0b247ae61\/v1.txt","buildinSkillList":[{"skillName":"utterances","skillId":"30f26aa6-4716-11ea-a2f1-d764e436f351","version":"latest"}],"buildinSkillRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","dictRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","skillRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","buildinDictList":{},"spkId":"30f26aa6-4716-11ea-a2f1-d764e436f351[latest]","env":"use_pinyin=1;use_custom_itn=1;vad_pause_time=1000;"},"mode":"LITE"},"recordId":"0561a525598711eca4e6e4b97a30e7cc","args":{"__internal__":"0","__auth__":"1","res":"aienglish-mix","productId":"12345_casr"},"eof":1,"result":{"res":"aicallcenter_cpu_v23","eof":1,"rec":"我 拿手 许嵩 的 歌","pinyin":"wo na shou xu song de ge"},"timelog":{"time":"2021-12-10T15:01:43+0800","message":{"realTime":[1639119703.873],"core":"ws:\/\/cn.asr.rec\/aitmp","startTime":1639119699.795,"feedNum":4,"resultTime":1639119703.873,"calcLastTime":1639119703.804,"calcStartTime":1639119699.797,"stopTime":1639119703.804,"ip":"10.12.7.141","lastTime":1639119702.805,"reqStartEndTime":1639119699.797,"calcTime":523.00000190735,"calcStartEndTime":1639119699.803,"feedTime":["1639119699.807,64001","1639119700.807,64001","1639119701.805,64001","1639119702.805,32001"]},"level":"INFO","tag":"OLIVE"},"serverIP":"10.12.8.30"}, client: 10.12.7.141, server: , request: "GET /runtime/v1/recognize?res=aienglish-mix&productId=12345_casr HTTP/1.1", host: "10.12.8.30:38002"

target server 接收到copy后服务请求的结果

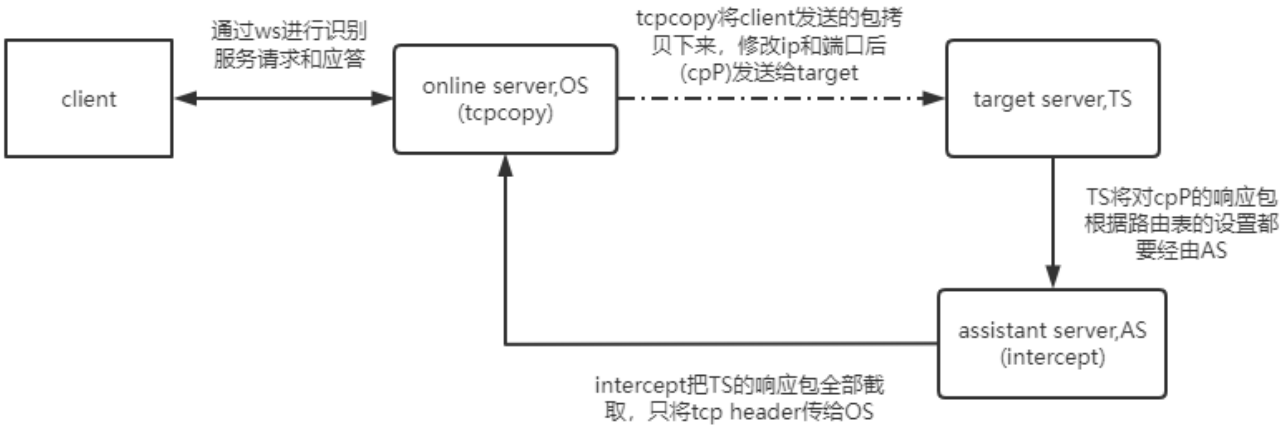
=aienglish-mix&productId=12345_casr HTTP/1.1", host: "10.12.8.30:38002"
2021/12/10 15:01:48 [info] 17#0: *5505 [lua] output.lua:0: result(): {"applicationId":"14796936228595de","source":"LITE","URL":"\/runtime\/v1\/recognize?res=aienglish-mix&productId=12345_casr","audioUrl":"http:\/\/records.unused.com\/0561a525598711eca4e6e4b97a30e7cc.wav","remoteIP":"172.16.58.83","time":"2021-12-10 15:01:48","params":{"app":{"applicationId":"14796936228595de","productId":"278583338"},"audio":{"sampleBytes":2,"audioType":"wav","sampleRate":16000,"channel":1},"request":{"skillList":[{"skillName":"utterances","skillId":"22372fce-4716-11ea-a43d-0f7f760adc83","version":"latest"}],"dictList":{},"coreType":"cn.asr.rec","customITNLexFile":"\/opt\/aispeech\/olive\/res\/asr_post_res\/test_itn\/custom_itn_lex.bin","buildinDictRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","res":"aitmp","recordId":"0561a525598711eca4e6e4b97a30e7cc","phraseFile":"\/opt\/aispeech\/olive\/res\/phrase_file_res\/48154546-8f69-11ea-a5e8-a7d0b247ae61\/v1.txt","buildinSkillList":[{"skillName":"utterances","skillId":"30f26aa6-4716-11ea-a2f1-d764e436f351","version":"latest"}],"buildinSkillRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","dictRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","skillRoot":"\/opt\/aispeech\/olive\/res\/lm_training_res\/","buildinDictList":{},"spkId":"30f26aa6-4716-11ea-a2f1-d764e436f351[latest]","env":"use_pinyin=1;use_custom_itn=1;vad_pause_time=1000;"},"mode":"LITE"},"recordId":"0561a525598711eca4e6e4b97a30e7cc","args":{"__internal__":"0","__auth__":"1","res":"aienglish-mix","productId":"12345_casr"},"eof":1,"result":{"res":"aicallcenter_cpu_v23","eof":1,"rec":"我 拿手 许嵩 的 歌","pinyin":"wo na shou xu song de ge"},"timelog":{"time":"2021-12-10T15:01:48+0800","message":{"realTime":[1639119708.706],"core":"ws:\/\/cn.asr.rec\/aitmp","startTime":1639119704.602,"feedNum":4,"resultTime":1639119708.706,"calcLastTime":1639119708.611,"calcStartTime":1639119704.605,"stopTime":1639119708.611,"ip":"172.16.58.83","lastTime":1639119707.612,"reqStartEndTime":1639119704.604,"calcTime":512.00008392334,"calcStartEndTime":1639119704.622,"feedTime":["1639119704.614,64001","1639119705.614,64001","1639119706.612,64001","1639119707.612,32001"]},"level":"INFO","tag":"OLIVE"},"serverIP":"10.12.8.30"}, client: 172.16.58.83, server: , request: "GET /runtime/v1/recognize?res=aienglish-mix&productId=12345_casr HTTP/1.1", host: "10.12.8.30:38002"
2021/12/10 15:01:48 [error] 17#0: *5505 [lua] prometheus.lua:0: log_error(): Unexpected nil value for label env of ngin

online server 接收原服务请求后的结果

2 问题小结

2.1 测试tcpcopy生效时，发现没有生效，target server没有识别脚本进行服务【阻塞时长：3 days】

排查过程：



- ping <IP>确认连通性，发现整个链路都已连通；（IP链路正常）
- nc -vz <IP:port>，ssh -v -p <port> <usr>@<IP>指令，检测端口8.30/38002，7.9/58002，发现都能正常接收数据；（端口正常）
- 布置websocket通讯项目到OS和TS上，发现tcpcopy能正常使用，websocket协议也能够兼容，只是在服务上不能正常使用；（tcpcopy正常）
- 执行脚本，TS和OS的服务都能正常运行；（服务正常）
- 最后通过tcpdump抓包发现，tcpdump -i <dev> port <PORT> 抓网口能抓到client→OS的数据包，但是只抓port时抓不到。tcpcopy默认方法里不要求提供网卡信息，即不抓网卡，由此推测是由于某原因导致从tcpcopy无法从IP层抓到client→OS的casr服务的数据包。（特定环境下，抓包异常）
- 调整编译运行参数（1.4.2中的-i参数，默认是关闭的，需要安装pcap的资源库，通过./config --pcap-capture激活），让tcpcopy能够在数据链路层抓包，发现问题解决。

原因：由于某原因导致从tcpcopy无法从IP层抓到client→OS的casr服务的数据包，猜测是虚拟主机的环境影响

解决方法：激活tcpcopy的链路层抓包功能（1.4.2中的-i参数，默认关闭，需要安装pcap的资源库，通过./config --pcap-capture激活）、

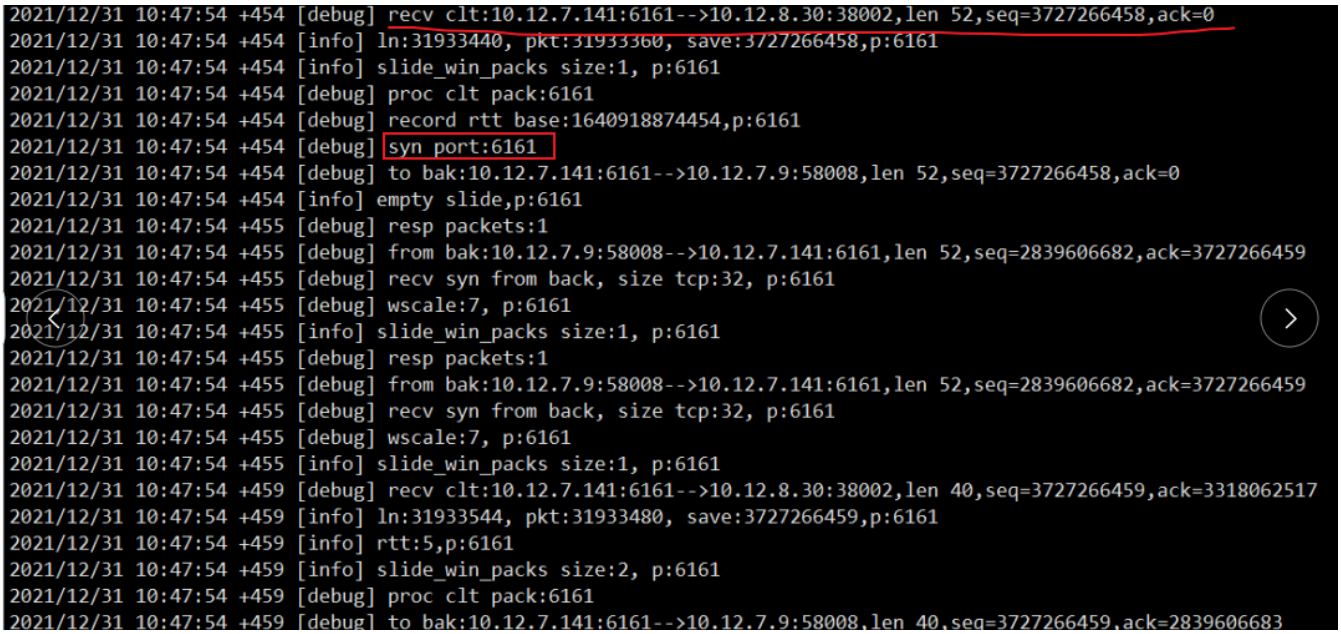
反思：

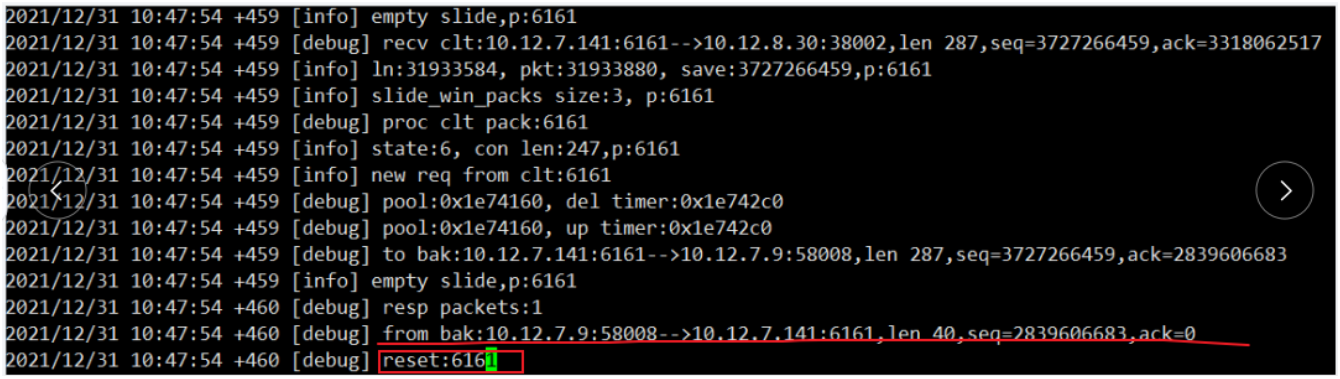
- 不能够结合实际灵活运用网络协议知识，无法规划出具体的排查流程；
- 对网络工具（特别是tcpdump）不熟悉，每个疑点的排查进度缓慢；
- 过分相信日志，定位问题时出错（认为intercept失效）。

2.2 tcpcopy拷贝到target失败，原因是产生第二次握手后target收到了reset

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.12.7.141	10.12.7.9	TCP	66	6161 → 58008 [SYN] Seq=0 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
2	0.000138	10.12.7.9	10.12.7.141	TCP	66	58008 → 6161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000539	10.12.7.141	10.12.7.9	TCP	60	6161 → 58008 [RST] Seq=1 Win=0 Len=0
4	0.005028	10.12.7.141	10.12.7.9	TCP	60	6161 → 58008 [ACK] Seq=1 Ack=1 Win=131072 Len=0
5	0.005095	10.12.7.141	10.12.7.9	HTTP	301	GET /runtime/v1/recognize?res=aienglish-mix&productId=12345_casr HTTP/1.1
6	0.005123	10.12.7.9	10.12.7.141	TCP	54	58008 → 6161 [RST] Seq=1 Win=0 Len=0
7	0.005144	10.12.7.9	10.12.7.141	TCP	54	58008 → 6161 [RST] Seq=1 Win=0 Len=0
8	1.018064	10.12.7.141	10.12.7.9	TCP	60	[TCP Port numbers reused] 6161 → 58008 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
9	1.018189	10.12.7.9	10.12.7.141	TCP	58	58008 → 6161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
10	1.019179	10.12.7.141	10.12.7.9	TCP	60	6161 → 58008 [RST] Seq=1 Win=0 Len=0
11	1.019251	10.12.7.141	10.12.7.9	TCP	60	6161 → 58008 [ACK] Seq=1 Ack=1 Win=65535 Len=0
12	1.019303	10.12.7.9	10.12.7.141	TCP	54	58008 → 6161 [RST] Seq=1 Win=0 Len=0
13	1.019321	10.12.7.141	10.12.7.9	TCP	361	6161 → 58008 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=307
14	1.019356	10.12.7.9	10.12.7.141	TCP	54	58008 → 6161 [RST] Seq=1 Win=0 Len=0
15	1.019400	10.12.7.141	10.12.7.9	TCP	1514	6161 → 58008 [ACK] Seq=308 Ack=1 Win=65535 Len=1460

日志排查：





可以看到tcpcopy从收到client的syn请求报文开始直到收到target的reset报文，从client接收reset或者自己发出faked reset报文，说明这条reset报文来自于非tcpcopy链路中的机器。

原因：tcpcopy配置时的faked ip被设置为target同网段的可达机器10.12.7.141，导致target的第二次握手报文被faked ip对应的真实机器接收。该机器未参与tcp握手流程，发送reset拒绝连接，target接收后将连接中断并发送reset报文。

解决方式：

1. faked ip修改为不可达的机器
2. 设置intercept机器的ip_forword为0，即关闭数据包转发功能

2.3 tcpdump抓包正常运行的tcpcopy，tcp连接发现在4次挥手时被reset，报文来源于tcpcopy的send_faked_reset()

tcpcopy发送faked reset有如下情况：

1. online server在第一次握手结束后，收到了target的第二次握手的没有syn标志的异常报文；
2. online server在第一次挥手结束后，收到了target的第二次挥手的没有fin标志的异常报文；
3. session连接超时；
4. client和online server的tcp连接正常流程结束，online发送reset给target快速释放连接；

原因：情况4，正常结束，快速释放连接；

2.4 tcpcopy在服务开启实时反馈的情况下，识别成功率降低

原因：开启实时反馈的过程中，服务会在识别过程中实时向客户端返回识别结果，而在不同服务上实时反馈不幂等，tcpcopy工具要求OS和TS上的服务接口幂等。tcpcopy认为应该接收到TS的TEXT报文的时候（根据clt接收的OS的TEXT报文时序）接收不到（接口不幂等，TS没发），停止发送数据报文，服务识别停止，导致整个连接阻塞。多并发下，大量挂起的session会占用服务的worker，最终导致识别率成功低。

解决方案：开启'-W'参数，tcpcopy不等待TS服务响应进行发包；但是直接开启，tcpcopy在完成发包后会不等待服务返回结果直接关闭连接，导致服务连接阻塞，识别不成功。修改源码，注释掉字段使得tcpcopy能够高发包优先级的同时在返回识别结果前保持住连接。

2.5 tcpcopy不通同时日志反馈[warn] many connections can't be established

解决方案：https://blog.csdn.net/longzhizhui926/article/details/100084956

tcpcopy部署——离线流量回放

1 部署流程

1.1 tcpcopy环境布置

tcpcopy布置到online server 10.12.8.30上

intercept布置到assistant server 10.12.7.8上

1.2 设置target server路由

set route

```
route add -net 10.12.7.0 netmask 255.255.255.0 gw 10.12.7.8
```

将target server(7.9)上所有发送到7.0网段上的包都路由到assistan server(7.8)上

1.3 配置/编译/启动intercept

1. assistant server需要关闭路由转发功能，echo 0 > /proc/sys/net/ipv4/ip_forward;
2. 在intercept路径下,顺序执行./configure <option> →make→ make clean，编译intercept
3. route指令查看路由表，找target server的ip的对应项的网卡；<pcap filter statment> 填tcp and src port 58002即可抓取target server(7.9:58002)的tcp响应包

run intercept

```
./usr/local/intercept/sbin/intercept -i <> -F '<pcap filter statment>' -
d
```

1.4 配置/编译/启动tcpcopy

1. 在tcpcopy路径下，按序执行./configure --offline（必需）→make→ make clean，编译tcpcopy；（注意，离线回放配置需要安装pcap库）
2. -i 指定离线流量文件.pcap文件（注意tcpdump监视抓取流量的端口必须和localServerPort保持一致），<transfer>格式为localServerPort-targetServerIP:targetServerPort，拷贝得到的包源IP修改为<fakedIP>，以此设置target server的路由；
3. tcpcopy运行必须在intercept之后，否则一定失败

run tcpcopy —offline

```
./usr/local/intercept/sbin/tcpcopy -x <transfer> -s <assistantServerIP>
-c <fakedIP> -i <pcap file>
```

2 问题小结

2.1 tcpcopy离线回放开启'-W'参数下无法正常完成识别，出现interruputed【阻塞时长：2d】

问题成因：TS机器的服务在完成识别之前，收到了被拷贝的报文websocket close，导致服务识别中断；根本原因是tcpcopy未能按照pcap文件的时序进行发包，此时close报文发包相较于源文件的时序更早。

解决方法：修改源码，使得tcpcopy能够严格按照pcap的文件时序发包，问题解决