## 上交所报盘协议解析文档

Macbook笔记本，位置在：~/pwork/sniff

世纪证券、银河证券

Tcpdump日志：sniff1.log

Hdf结构化日志：sniff.h5

解析代码：showsniff.py

参考文档：《前后台消息接口协议分析\_V1.0.33.pdf》

代理中介：middleman.py

代理链工具：proxychains

报盘日志：ud.log

解析代码：showprox.py

1. 通信连接

世纪证券

报盘机：220.82.46.15

上交所网关：180.2.176.66

银河证券

报盘机：n/a

上交所网关：42.24.0.06

1. 心跳消息

消息体长度：84

特征：可显示字串中出现9C1\_15

频率：每4秒一条

模式：一条请求、一条响应

消息格式：BeNetMsgReqRspT

数据格式：不明

1. 对时消息

消息体长度：153

特征：无

频率：每20秒一条（如果多个PBU，则间隔1秒依次对时）

模式：一条响应

消息格式：NetPackageT

数据格式：roesgtscT

[111:135]分别为OES发送时间、CS接收时间、CS发送时间

1. 短订单消息

消息体长度：113

特征：可显示字串中出现PBU

频率：数目较多、无固定频率

模式：一条请求、一条响应

消息格式：NetPackageT

数据格式：VSBeNetMsgReqRspT

[142:182]为应用头，其中setId、msgId字段能够对上

1. Tcpdump抓包

启动抓包工具

sudo tcpdump host nnn.nnn.nnn.nnn -i eth0 -nnevv -XX > sniff.log

其中nnn.nnn.nnn.nnn 用EzCS IP地址替换

转换程序：sniff.py

将非结构化日志转换成为结构化的hdf文件（sniff.h5）

1. 模拟连接

服务端（AC1主机：47.100.110.38）

nc -lk 8888

客户端（NC5主机：1.94.4.53）

nc a1.hifiax.com 8888

1. 拦截工具

~~ProxyChains可将单个程序的所有TCP连接都导向代理，但是不区分去往哪个IP、哪个端口~~

Iptables可以将特定IP、特定端口的流量重定向到本地代理

sudo iptables -t nat -A OUTPUT -p tcp -d 47.100.110.38 -m owner --uid-owner uproxy -j RETURN

sudo iptables -t nat -A OUTPUT -p tcp -d 47.100.110.38 --dport 10006 -j DNAT --to-destination 127.0.0.1:10006

sudo iptables -t nat -A OUTPUT -p tcp -d 47.100.110.38 --dport 10005 -j DNAT --to-destination 127.0.0.1:10005

sudo iptables -t nat -A OUTPUT -p tcp -d 47.100.110.38 --dport 10003 -j DNAT --to-destination 127.0.0.1:10003

查看路由表

sudo iptables -t nat -L -v

服务器重启后，路由表自动恢复

或者输入以下命令，恢复路由表

sudo iptables -t nat -D OUTPUT -p tcp -d 47.100.110.38 --dport 10006 -j DNAT --to-destination 127.0.0.1:10006

sudo iptables -t nat -D OUTPUT -p tcp -d 47.100.110.38 --dport 10005 -j DNAT --to-destination 127.0.0.1:10005

sudo iptables -t nat -D OUTPUT -p tcp -d 47.100.110.38 --dport 10003 -j DNAT --to-destination 127.0.0.1:10003

sudo iptables -t nat -D OUTPUT -p tcp -d 47.100.110.38 -m owner --uid-owner uproxy -j RETURN

1. 代理协议（弃用）

下面是客户端向SOCKS 4代理服务器，发送的连接请求包的格式（以字节为单位）：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VN | CD | DSTPORT | DSTIP | USERID | NULL |
| 1 | 1 | 2 | 4 | variable | 1 |

* VN是SOCK版本，应该是4；
* CD是SOCK的命令码，1表示CONNECT请求，2表示BIND请求；
* DSTPORT表示目的主机的端口；
* DSTIP指目的主机的IP地址；
* NULL是0；

代理服务器而后发送回应包（以字节为单位）：

|  |  |  |  |
| --- | --- | --- | --- |
| VN | CD | DSTPORT | DSTIP |
| 1 | 1 | 2 | 4 |

* VN是回应码的版本，应该是0；
* CD是代理服务器答复，有几种可能：
  + 90，请求得到允许；
  + 91，请求被拒绝或失败；
  + 92，由于SOCKS服务器无法连接到客户端的identd（一个验证身份的进程），请求被拒绝；
  + 93，由于客户端程序与identd报告的用户身份不同，连接被拒绝。
* DSTPORT与DSTIP与请求包中的内容相同，但被忽略。

1. 代理连接（单端口）

服务端（AC1主机：47.100.110.38）

nc -lk 10006

客户端代理（NC5主机：1.94.4.53）

python middleman.py

检查

netstat -apn | grep 10006

客户端（NC5主机：1.94.4.53）

proxychains4 nc 47.100.110.38 10006

1. 代理连接（多端口）

回声服务端代码：echo.py

服务端日志：echo.log

服务端（AC1主机：47.100.110.38）

bash nlisten.sh （监听10006、10005、10003端口）

检查

ps -ef | grep echo

~~ps -ef | grep nc~~

客户端代理（NC5主机：1.94.4.53）

~~python middleman.py~~

python middleguy.py

检查

netstat -apn | grep 10006

报盘客户端模拟代码：simoes.py

客户端（NC5主机：1.94.4.53）

./simoes.py （根据命令行输入，向服务端的10006、10005、10003分别发送消息）

~~proxychains4 ./simoes.py （根据命令行输入，向服务端的10006、10005、10003分别发送消息）~~

~~proxychains4 bash nchat.sh （向47.100.110.38的10006、10005、10003分别发送消息）~~

1. middleman代理程序（弃用）

server：监听端口，接受连接后转为clients

clients：连接后的首条消息，也就是代理消息，处理首条后转为ud/md/rfclients

forwarders：转发至交易所端的UD连接，以及其它连接

mdforwarders：转发至交易所端的MD连接

rfforwarders：转发至交易所端的RF连接

udclients：本地拦截来的UD连接，从clients代理分流而来

mdclients：本地拦截来的MD连接，从clients代理分流而来

rfclients：本地拦截来的RF连接，从clients代理分流而来

注意要用select处理网络消息，对端close会触发一条空消息

备注：原先为middleman.py，用一个端口监听三个端口，要处理PROX代理消息，现弃用

1. middleguy代理程序

server：监听端口，接受连接后转为clients

clients：本地拦截来的UD连接

mdclients：本地拦截来的MD连接

rfclients：本地拦截来的RF连接

forwarders：转发至交易所端的UD连接

mdforwarders：转发至交易所端的MD连接

rfforwarders：转发至交易所端的RF连接

注意要用select处理网络消息，对端close会触发一条空消息

1. 查看日志

python showprox.py

~~xxd ud.log~~