OpenVPN服务搭建与管理

引言：

本文利用OpenVPN搭建VPN服务，并利用pam_sqlite3插件实现用户认证；通过openvpn_web进行用户管理与日志系统。

一、安装OpenVPN服务

基础环境：

服务端： CentOS 7.6

客户端：Windows 7

OpenVPN: openvpn-2.4.7 (https://github.com/OpenVPN/openvpn)

easy-rsa：easy-rsa 3.0.6 (https://github.com/OpenVPN/easy-rsa)

OpenVPN GUI: openvpn gui (https://gitee.com/lang13002/openvpn-portable)

服务器：

外网IP： 1.10.10.123 ens33

内网IP： 192.168.100.100 ens34

1.1 安装openvpn

安装依赖包

```
# yum install -y gcc gcc-c++ libtool automake lz4-devel lzo-devel pam-devel openssl-devel systemd-devel sqlite-devel
```

从github上下载openvpn源代码包并解压

```
# wget https://github.com/OpenVPN/openvpn/archive/v2.4.7.tar.gz
# tar -xvf v2.4.7.tar.gz
```

编译openvpn并安装

```
# cd openvpn-2.4.7
# autoreconf -i -v -f
# ./configure --prefix=/usr/local/openvpn --enable-lzo --enable-lz4 --enable-crypto --enable-server --enable-plugins --enable-port-share --enable-iproute2 --enable-pf --enable-plugin-auth-pam --enable-pam-dlopen --enable-systemd
# make && make install
```

配置系统服务

安装openvpn-server@.service系统服务

```
# cp /usr/local/openvpn/lib/systemd/system/openvpn-server@.service
/usr/lib/systemd/system/
```

设置openvpn-server@.service启动

```
# ln -s /usr/lib/systemd/system/openvpn-server\@.service
/etc/systemd/system/multi-user.target.wants/openvpn-server@server.service
# systemctl daemon-reload
# systemctl enable openvpn-server@server
```

1.2 生成证书

下载easy-rsa3并解压

```
# wget https://github.com/OpenVPN/easy-rsa/archive/v3.0.6.tar.gz
# tar -xvf v3.0.6.tar.gz
```

根据easy-rsa-3.0.6/easyrsa3/vars.example文件生成全局配置文件vars

```
# cd easy-rsa-3.0.6/easyrsa3/
# cp vars.samples vars
```

修改vars文件，根据需要去掉注释，并修改对应值

```
set_var EASYRSA_REQ_COUNTRY     "CN"
set_var EASYRSA_REQ_PROVINCE    "HUBEI"
set_var EASYRSA_REQ_CITY        "WUHAN"
set_var EASYRSA_REQ_ORG "ZJ"
set_var EASYRSA_REQ_EMAIL       "zj@test.com"
set_var EASYRSA_REQ_OU          "ZJ"
set_var EASYRSA_KEY_SIZE        2048
set_var EASYRSA_ALGO            rsa
set_var EASYRSA_CA_EXPIRE       3650
```

生成服务端证书

```
# ./easyrsa init-pki     # 初始化，生成一系列文件与目录
# ./easyrsa build-ca     # 生成根证书，记住ca密码
# ./easyrsa build-server-full server nopass # 生成服务端证书，nopass参数生成一个无密码的证书
# ./easyrsa gen-dh       # 生成Diffie-Hellman
```

生成客户端证书

```
# ./easy-rsa build-client-full client1 nopass
注：可生成client1, client2, client3或对应姓名的客户端证书
```

为了提高安全性，生成ta.key

```
# openvpn --genkey --secret ta.key
```

整理服务端证书

```
# cp pki/ca.crt /etc/openvpn/server/
# cp pki/private/server.key /etc/openvpn/server/
# cp pki/issued/server.crt /etc/openvpn/server/
# cp pki/dh.pem /etc/openvpn/server/
# cp ta.key /etc/openvpn/server/
```

1.3 添加SQLite认证

下载pam_sqlite3并安装

```
# git clone https://gitee.com/lang13002/pam_sqlite3.git
# cd pam_sqlite3
# make && make install
```

添加pam认证文件

```
# vim /etc/pam.d/openvpn
auth        required    pam_sqlite3.so db=/etc/openvpn/openvpn.db table=t_user user=username passwd=password expire=expire crypt=1
account     required    pam_sqlite3.so db=/etc/openvpn/openvpn.db table=t_user user=username passwd=password expire=expire crypt=1
```

导入sqlite3数据库文件，创建数据库

```
# sqlite3 /etc/openvpn/openvpn.db
sqlite> .read openvpn_web/doc/openvpn.sql
```

1.4 创建服务端配置文件

参照sample/sample-config-files/server.conf文件

```
# vim /etc/openvpn/server/server.conf
port 1194
proto tcp-server
;proto udp
dev tun
topology subnet

ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh /etc/openvpn/server/dh.pem

cipher AES-256-CBC
auth SHA512
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-256-CBC-SHA:TLS-DHE-RSA-WITH-CAMELLIA-256-CBC-SHA:TLS-DHE-RSA-WITH-AES-128-CBC-SHA:TLS-DHE-RSA-WITH-CAMELLIA-128-CBC-SHA

tls-auth /etc/openvpn/server/ta.key 0
;tls-crypt /etc/openvpn/server/ta.key    ;客户端将ta.key嵌入到配置文件

user openvpn
group openvpn
```

```
server 10.8.0.0 255.255.255.0
push "dhcp-option DNS 114.114.114.114"
push "route 192.168.100.0 255.255.255.0"
push "route-gateway 192.168.100.1"

verify-client-cert require
username-as-common-name
plugin /usr/local/openvpn/lib/openvpn/plugins/openvpn-plugin-auth-pam.so openvpn

keepalive 10 120
comp-lzo
compress "lz4"
persist-key
persist-tun
status /var/log/openvpn-status.log
log    /var/log/openvpn.log
verb 3
```

1.5 开启路由转发功能

```
# 路由转发
# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1

# 临时启用
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

1.6 启动openvpn服务

```
# systemctl start openvpn-server@server
```

二、客户端配置

2.1 下载客户端程序：

从https://gitee.com/lang13002/openvpn-portable/repository/archive/v1.0下载程序，并安装网卡驱动；

2.2 安装驱动：

运行openvpn-portable/tap-windows.exe

2.3 设置客户端证书

将上面生成的ca.crt, client1.crt, client1.key放到openvpn-portable的data/config下，并修改客户端配置

```
client
dev tun
proto tcp-client
remote vpnserver.com 1194

allow-recursive-routing
```

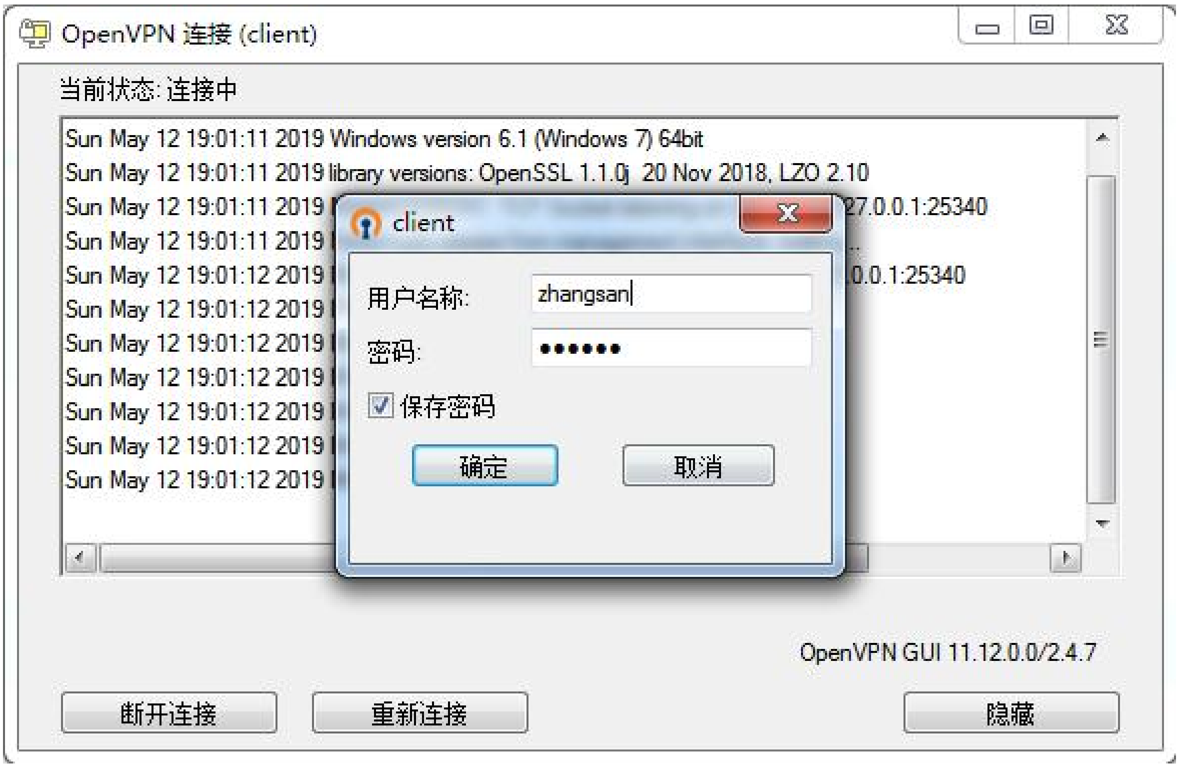
```
resolv-retry infinite
nobind
persist-key
persist-tun

remote-cert-tls server
auth-user-pass
auth-nocache
ca ca.crt
cert client1.crt
key client1.key

cipher AES-256-CBC
auth SHA512
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-256-CBC-SHA:TLS-DHE-RSA-WITH-CAMELLIA-256-CBC-SHA:TLS-DHE-RSA-WITH-AES-128-CBC-SHA:TLS-DHE-RSA-WITH-CAMELLIA-128-CBC-SHA

tls-auth ta.key 1

comp-lzo
compress lz4
verb 3
mute 20
```

```
注：当有多个客户端时，有多个文件(ca.crt，client1.crt，client1.key，client.ovpn)需要分发给客户，势必会很麻烦；可以将证书嵌入到客户端配置文件中；
;ca ca.crt         // 将这行注释掉
;cert client.crt   // 将这行注释掉
;key client.key    // 将这行注释掉
;tls-auth ta.key 1 // 将这行注释掉
<ca>
-----BEGIN CERTIFICATE-----
MIIDGDCCAgCgAwIBAgIJAI9Ld4PlKEiOMA0GCSqGSIb3DQEBCwUAMA0xCzAJBgNV
....
OCeTQvQ4WhyIvVgURV3ITcAKYFKUQ1sPbpjuZg==
-----END CERTIFICATE---
</ca>
<cert>
-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIRAIZoEQ5PvHDs9xpTLMP3RqMwDQYJKoZIhvcNAQELBQAw
......
nCpzC3l8sVezxk2r
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDw1iq3HBe1otCU
......
ullaNc6mu3N/wTPZoQhDOKAO
-----END PRIVATE KEY-----
</key>
<tls-crypt>
#
# 2048 bit OpenVPN static key
```

```
#
-----BEGIN OpenVPN Static key V1-----
376ff00121bc6cd39fe1382c44be1433
......
-----END OpenVPN Static key V1-----
</tls-crypt>
```

2.4 连接VPN

启动openvpn-porable



OpenVPN 连接 (client)

当前状态: 连接中

```
Sun May 12 19:01:11 2019 Windows version 6.1 (Windows 7) 64bit
Sun May 12 19:01:11 2019 library versions: OpenSSL 1.1.0j  20 Nov 2018, LZO 2.10
Sun May 12 19:01:11 2019                                  27.0.0.1:25340
Sun May 12 19:01:11 2019                                  ...
Sun May 12 19:01:12 2019                                  0.0.1:25340
Sun May 12 19:01:12 2019
Sun May 12 19:01:12 2019
Sun May 12 19:01:12 2019
Sun May 12 19:01:12 2019
Sun May 12 19:01:12 2019
Sun May 12 19:01:12 2019
```

client

用户名称: zhangsan

密码: •••••••

☑ 保存密码

三、OpenVPN用户管理与日志

3.1 安装依赖

```
# pip install peewee tornado pycryptodome apscheduler
```

3.2 下载openvpn-web

```
# git clone https://gitee.com/lang13002/openvpn_web.git
```

3.3 添加日志脚本

服务端配置添加运行脚本

```
script-security 2
client-connect /etc/openvpn/server/connect.py
client-disconnect /etc/openvpn/server/disconnect.py
```

connect.py

```
#!/usr/bin/python

import os
import time
import sqlite3

username = os.environ['common_name']
trusted_ip = os.environ['trusted_ip']
trusted_port = os.environ['trusted_port']
local = os.environ['ifconfig_local']
remote = os.environ['ifconfig_pool_remote_ip']
timeunix= os.environ['time_unix']

logintime = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(time.time()))

conn = sqlite3.connect("/etc/openvpn/openvpn.db")
cursor = conn.cursor()
query = "insert into t_logs(username, timeunix, trusted_ip, trusted_port, local,
remote, logintime) values('%s','%s', '%s', '%s', '%s', '%s', '%s')" %
(username, timeunix, trusted_ip, trusted_port, local, remote, logintime)
cursor.execute(query)
conn.commit()
conn.close()
```

disconnect.py

```
#!/usr/bin/python

import os
import time
import sqlite3

username = os.environ['common_name']
trusted_ip = os.environ['trusted_ip']
received = os.environ['bytes_received']
sent = os.environ['bytes_sent']

logouttime = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(time.time()))

conn = sqlite3.connect("/etc/openvpn/openvpn.db")
cursor = conn.cursor()
query = "update t_logs set logouttime='%s', received='%s', sent= '%s' where
username = '%s' and trusted_ip = '%s'" %  (logouttime, received, sent, username,
trusted_ip)
cursor.execute(query)
conn.commit()
conn.close()
```

3.4 初始化配置

使用OpenSSL生成密钥对

```
# openssl genrsa -out private.pem 1024
# openssl rsa -in private.pem -pubout -out public.pem
# cat private.pem
# cat public.pem
```

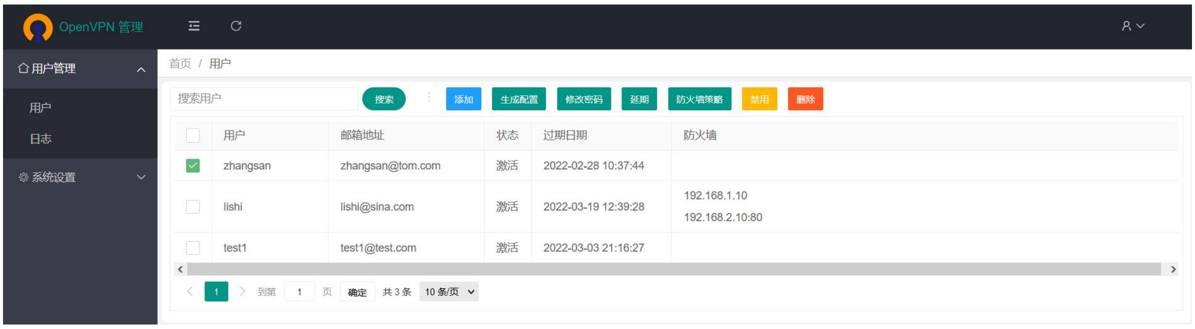
修改config.py文件，指定sqlite3 数据文件位置及用于用户密码加密与解密的密钥对

```
dbfile="/etc/openvpn/openvpn.db"
private_key=b'-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAqif+G/cpiP582c2JGkA6cb8eIrzUUq9eF1swwhVRjKcEMaQz\nQzH01pxb4GJT
PPNl64YK2uyZWKDwvrwhnas2v4GzpXNL1tKv4YYlT3rSvcFF3ouw\nTICsOvZaaZso7w7W2NzvvF0vGO
Wv7o8aD+hBXlcZUVaXISaodQW8+aMNs0GMVwmg\nXEVtGsjg+LuoDDiRnkq3B0lClAroa1tzMw2yAu+g
urYzEaZ1rt4JOzo4RMwu+CqI\ninqbIPGLzZ71CWhdlA6AKKzTOjfgEnzQDkNrzWpSAPIK6GaIwOSRbw
8JswhuBVOI\nNGsjC5lIg5U8hOWKZGIWqwYC58uhnDW3b8BdNwIDAQABAoIBABC0X2gboi2Lg/J8\nxD
gNbfXRXbu3gdKvDVPJ3vLW1YwzSbnQrtWq147Wh9byxpnMii90/kDS1664Ehpv\nhsKFvTJQB4Gd9ltb
9x2/v47OQspgaVcS3wpyDjaCc/E7mDYSS1/vHlWfaLdb6TzI\nbYMTrPZkvGmo6hC8ZlWJH2D7AHMEbb
qD4SuYdilEmejOu7Ec1rngFFuVD1xY0LeQ\nQlEwXWwijSqI3E5o/XyRZGvkft4d7seMxmswfVFXxlK7
JX/hPV0qqk9AjaBQJK2u\n1wZUqVcRalrZWGa4pK7VaqabC3qtFLPL0uLaJ2elW8dMhtjwHts50fqpi0
5P6FWJ\nzMXCxNECgYEAvxZAnAjUlcFRQE4mx0oMmQ/wdYXq9SX6KSfHgSci8yMRi25JovhG\nJhxf4P
adbhYCWfvg/G5/XDJjzLH+Q2xhMCCMilKG9hC8xja+wtnwijdWAYUUhhcm\nWvHKNG4dTdEkY93EOGlW
slbPsRJ1+uE9+jeMRdqfOpKkc01zVcg+DpECgYEA4/WD\n/bjPgWlYPiTTCGMBsXzpp618aMXYbUh3CR
AORuXiethfq2dW4ogPl2RbCyqC1AX2\nPOLoTEORVDKuzEFWDdKOFskI7A1xTiPwccDCq5RCPwWSZxaT
5DAY5Ec8cl/5/P9E\np+VEfMtPRDt6YDsLwbcCHegLF+7IYpnP+RkEoOcCgYBdJHKf3CoLSSzaxH1gfP
bB\nESTrlDhgAH/82ZgEm1gM3dYqebrJBm3jG8ecd3lrdKz2wbD8Orw365P77fL7WHPT\nKrp2nh2NCc
GKeJrpjaQTKz/wA7dqWRRoFh2zCs2b5crwJuQDf000Jt6b+FyrymkU\nc0kbr6IXwe0JCwKiGLYvEQKB
gQC9UJmkbgZOyEkTmwtzrJ2sZDu8GHT4ok5iO8s3\nyJDCyonzUZzqQXFDwpGIPjzqIgzyvlzIJf2b0I
VyMoE+eohYBGQigiSZvXQ629gE\n8Hv7eK4nnp3+ZR6/ZD5X3t1Rc2mudeTztpDRPxt+ZBL2tjLGVxE3
+wyzfIgIcwro\nKaHTYwKBgEt8wiU1zjnXFdBWFBQ9ldqkZ4YgoUS6yiFzy8aPwR3faTEG68aGS9lC\n
whdatQmT4vprCASCAELGNJD7DP8/nxTtF6oVBg69LSsxv2gowduw1u1sg56ytMOc\nuRTbuKdblaDS65
LDIbz7c5nVEJ+ZGUL88bbPxZ+sWXEWFRzGZNj+\n-----END RSA PRIVATE KEY-----'
public_key=b'-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqif+G/cpiP582c2JGkA6\ncb8eIrzUUq9e
F1swwhVRjKcEMaQzQzH01pxb4GJTPPNl64YK2uyZWKDwvrwhnas2\nv4GzpXNL1tKv4YYlT3rSvcFF3o
uwTICsOvZaaZso7w7W2NzvvF0vGOWv7o8aD+hB\nXlcZUVaXISaodQW8+aMNs0GMVwmgXEVtGsjg+Luo
DDiRnkq3B0lClAroa1tzMw2y\nAu+gurYzEaZ1rt4JOzo4RMwu+CqIinqbIPGLzZ71CWhdlA6AKKzTOj
fgEnzQDkNr\nzWpSAPIK6GaIwOSRbw8JswhuBVOINGsjC5lIg5U8hOWKZGIWqwYC58uhnDW3b8Bd\nNw
IDAQAB\n-----END PUBLIC KEY-----'
```

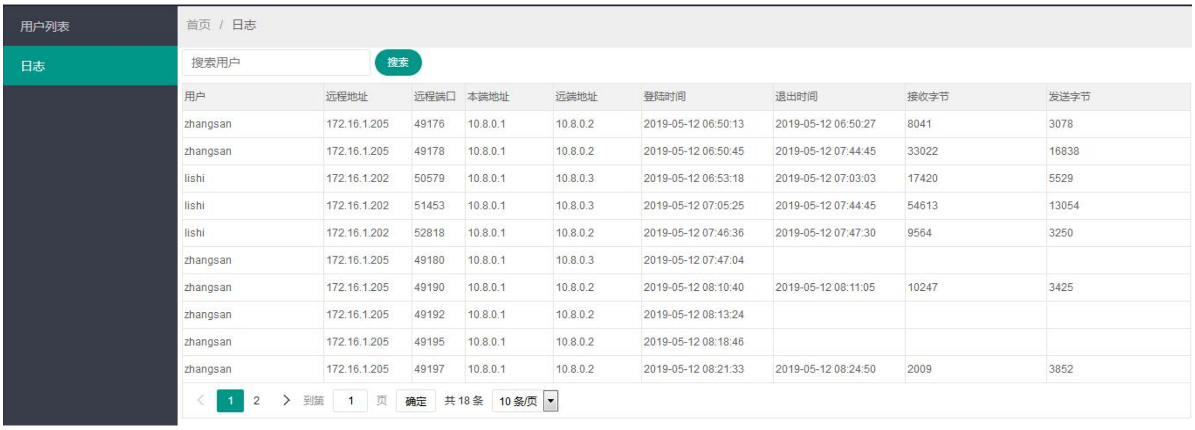
3.5 启动服务

```
# python myapp.py
```

3.6 管理界面



□	用户	邮箱地址	状态	过期日期	防火墙
☑	zhangsan	zhangsan@tom.com	激活	2022-02-28 10:37:44	
□	lishi	lishi@sina.com	激活	2022-03-19 12:39:28	192.168.1.10 192.168.2.10:80
□	test1	test1@test.com	激活	2022-03-03 21:16:27	



用户	远程地址	远程端口	本端地址	远端地址	登陆时间	退出时间	接收字节	发送字节
zhangsan	172.16.1.205	49176	10.8.0.1	10.8.0.2	2019-05-12 06:50:13	2019-05-12 06:50:27	8041	3078
zhangsan	172.16.1.205	49178	10.8.0.1	10.8.0.2	2019-05-12 06:50:45	2019-05-12 07:44:45	33022	16838
lishi	172.16.1.202	50579	10.8.0.1	10.8.0.3	2019-05-12 06:53:18	2019-05-12 07:03:03	17420	5529
lishi	172.16.1.202	51453	10.8.0.1	10.8.0.3	2019-05-12 07:05:25	2019-05-12 07:44:45	54613	13054
lishi	172.16.1.202	52818	10.8.0.1	10.8.0.2	2019-05-12 07:46:36	2019-05-12 07:47:30	9564	3250
zhangsan	172.16.1.205	49180	10.8.0.1	10.8.0.3	2019-05-12 07:47:04			
zhangsan	172.16.1.205	49190	10.8.0.1	10.8.0.2	2019-05-12 08:10:40	2019-05-12 08:11:05	10247	3425
zhangsan	172.16.1.205	49192	10.8.0.1	10.8.0.2	2019-05-12 08:13:24			
zhangsan	172.16.1.205	49195	10.8.0.1	10.8.0.2	2019-05-12 08:18:46			
zhangsan	172.16.1.205	49197	10.8.0.1	10.8.0.2	2019-05-12 08:21:33	2019-05-12 08:24:50	2009	3852