日志的种类和记录的方式-自定义ssh服务日志类型和存储位置

在centos7中，系统日志消息由两个服务负责处理：systemd-journald和rsyslog

1.常见日志文件的作用

系统日志文件概述：/var/log目录保管由rsyslog维护的，里面存放的一些特定于系统和服务的日志文件

日志文件	用途
/var/log/message	大多数系统日志消息记录在此处。也有例外的：如与身份验证，电子邮件处理相关的定期作业任务等
/var/log/secure	安全和身份验证相关的消息和登录失败的日志文件。ssh远程连接产生的日志
/var/log/maillog	与邮件服务器相关的消息日志文件
/var/log/cron	与定期执行任务相关的日志文件
/var/log/boot.log	与系统启动相关的消息记录
/var/log/dmesg	与系统启动相关的消息记录，一般记录内核

例1：查看哪个IP地址经常暴力破解系统用户密码

```
[root@exercise1 ~]# ssh root@192.168.245.169 #故意输错3次密码
[root@exercise2 ~]# grep Failed /var/log/secure
[root@exercise2 ~]# grep Failed /var/log/secure
Aug  2 14:58:15 home2 sshd[1076]: Failed password for root from 192.168.245.167 port 51142 ssh2
Aug  2 14:58:25 home2 sshd[1076]: Failed password for root from 192.168.245.167 port 51142 ssh2
Aug  2 14:58:30 home2 sshd[1076]: Failed password for root from 192.168.245.167 port 51142 ssh2
```

例2：/var/log/wtmp文件的作用

/var/log/wtmp也是一个二进制文件，显示到目前为止，成功登录系统的每个用户的登录次数和持续时间等信息。

可以用last命令输出wtmp中内容：last

```
[root@exercise1 ~]# last
root     pts/0        192.168.119.1    Fri Jan 28 15:10   still logged in
reboot   system boot  3.10.0-693.el7.x Fri Jan 28 14:44 - 15:16  (00:32)
……

或者

[root@exercise1 ~]# last -f /var/log/wtmp
```

```
[root@exercise1 ~]# last
root     pts/0        192.168.119.1    Fri Jan 28 15:10   still logged in
reboot   system boot  3.10.0-693.el7.x Fri Jan 28 14:44 - 15:16  (00:32)
root     pts/1        192.168.119.1    Sat Jan 29 02:03 - down   (00:59)
root     tty1                          Sat Jan 29 02:01 - 03:02  (01:00)
root     pts/0        192.168.119.1    Thu Jan 27 14:22 - down  (1+12:39)
reboot   system boot  3.10.0-693.el7.x Thu Jan 27 14:20 - 03:02 (1+12:41)
root     tty1                          Thu Jan 27 13:55 - 14:18 (1+00:23)
root     pts/2        192.168.119.1    Thu Jan 27 13:55 - down  (1+00:23)
root     pts/1        192.168.119.1    Thu Jan 27 12:45 - down  (1+01:33)
root     pts/0        192.168.119.1    Thu Jan 27 11:26 - down  (1+02:52)
reboot   system boot  3.10.0-693.el7.x Thu Jan 27 11:26 - 14:18 (1+02:52)
root     pts/0        192.168.119.1    Thu Jan 27 10:40 - crash  (00:46)
reboot   system boot  3.10.0-693.el7.x Thu Jan 27 10:39 - 14:18 (1+03:39)
root     pts/0        192.168.119.1    Wed Jan 26 21:58 - down   (01:19)
reboot   system boot  3.10.0-693.el7.x Wed Jan 26 21:57 - 23:18  (01:20)
root     pts/0        192.168.119.1    Thu Jan 20 19:20 - down   (01:25)
reboot   system boot  3.10.0-693.el7.x Thu Jan 20 19:20 - 20:45  (01:25)
root     pts/0        192.168.119.1    Wed Jan 19 22:27 - crash  (20:52)
reboot   system boot  3.10.0-693.el7.x Wed Jan 19 22:21 - 20:45  (22:24)
root     pts/1        192.168.119.1    Wed Jan 19 19:11 - 19:15  (00:03)
root     pts/0        192.168.119.1    Wed Jan 19 19:11 - down   (02:07)
reboot   system boot  3.10.0-693.el7.x Wed Jan 19 18:36 - 21:18  (02:42)
root     pts/1        192.168.119.1    Tue Jan 18 23:28 - down   (00:22)
root     pts/0        192.168.119.1    Tue Jan 18 22:34 - down   (01:17)
reboot   system boot  3.10.0-693.el7.x Tue Jan 18 22:27 - 23:51  (01:24)
root     pts/1        192.168.119.1    Tue Jan 18 19:35 - down   (01:00)
root     pts/0        192.168.119.1    Tue Jan 18 18:35 - down   (02:00)
reboot   system boot  3.10.0-693.el7.x Tue Jan 18 18:35 - 20:36  (02:00)
root     pts/0        192.168.119.1    Mon Jan 17 23:35 - crash  (18:59)
```

例3：使用/var/log/btmp文件查看暴力破解系统的用户

/var/log/btmp文件是记录错误登录系统的日志。如果发现/var/log/btmp日志文件比较大，大于1M，就算大了，就说明很多人在暴力破解ssh服务，此日志需要使用lastb程序查看

```
[root@exercise2 ~]# lastb
[root@exercise1 ~]# lastb
root     ssh:notty    192.168.245.167  Mon Aug  2 14:58
- 14:58  (00:00)
root     ssh:notty    192.168.245.167  Mon Aug  2 14:58
- 14:58  (00:00)
root     ssh:notty    192.168.245.167  Mon Aug  2 14:58
- 14:58  (00:00)
root     ssh:notty    192.168.245.1    Mon Aug  2 14:42
- 14:42  (00:00)
root     ssh:notty    192.168.245.1    Mon Aug  2 14:39
- 14:39  (00:00)
root     tty1                          Tue Jul 13 09:30
- 09:30  (00:00)
```

清空日志：

方法1：[root@base ~]#>/var/log/btmp

方法2：rm -rf /var/log/btmp && touch /var/log/btmp

两者的区别?

使用方法2，因为创建了新的文件，而正在运行的服务，还用着原来文件的inode号和文件描述码，所需要重启一下rsyslog服务。

建议使用方法1>/var/log/btmp

2.日志的记录方式

2.1、日志的分类:

```
daemon          后台进程相关
kern            内核产生的信息
lpr             打印系统产生的
authpriv        安全认证
cron            定时相关
mail            邮件相关
syslog          日志服务本身的
news            新闻系统
local0~7        自定义的日志设备
local0-local7   8个系统保留的类，供其它的程序使用或者是用户自定义
```

编码	优先级	严重性
8	none	不记录
7	debug	信息对开发人员调试应用程序有用，在操作过程中无用
6	info	正常的操作信息，可以收集报告，测量吞吐量等
5	notice	注意，正常但重要的事件
4	warning	警告，提示如果不采取行动。将会发生错误。比如文件系统使用90%
3	err	错误，阻止某个模块或程序的功能不能正常使用
2	crit	关键的错误，已经影响了整个系统或软件不能正常工作的信息
1	alert	警报,需要立刻修改的信息
0	emerg	紧急，内核崩溃等严重信息

3.rsyslog日志服务

cetos7>rsyslog日志服务>配置文件　/etc/rsyslog.conf

我们来查看一下日志的配置文件信息：

编辑配置文件vim /etc/rsyslog.conf

```
# Logging much else clutters up the screen.
kern.*                                                  /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

注释：

#kern.*内核类型的所有级别日志存放到 /dev/console

*.info;mail.none;authpriv.none;cron.none /var/log/messages

#所有的类别级别是info以上　　除了mail,authpriv,cron(产生的日志太多,不易于查看)

authpriv.*　　认证的信息->存放-> /var/log/secure

mail.*　　邮件相关的信息->存放-> -/var/log/maillog

cron.*　　计划任务相关的信息->存放 /var/log/cron

local7.*　　开机时显示的信息->存放--> /var/log/boot.log

注：1.举例　*.*　（相当于分类.级别）　:第一个*代表日志的分类，第二个*代表日志的优先级

2.“- ”号：　邮件的信息比较多,现将数据存储到内存,达到一定大小,全部写到硬盘.有利于减少 I/O 进程的开销

数据存储在内存,如果关机不当数据消失

4.日志输入的规则

.info　　大于等于info级别的信息全部记录到某个文件

.=级别　　仅记录等于某个级别的日志

.=info　　只记录info级别的日志

.!级别　　除了某个级别意外,记录所有的级别信息

例:　.!err除了err外记录所有

xxx.none指的是排除某个类别
例：mail.none　　所有mail类别的日志都不记录

5.实战-自定义ssh服务的日志类型和存储位置

```
[root@exercise1 ~]# vim /etc/rsyslog.conf    #以73行下，插入以下红色标记内容
```

73　　local7.*　　/var/log/boot.log
74　　local0.*　　/var/log/sshd.log
注：把local0类别的日志，保存到/var/log/sshd.log路径

```
72 # Save boot messages also to boot.log
73 local7.*                                                /var/log/boot.log
74
75
```

5.1、定义ssh服务的日志类别为local0，编辑sshd服务的主配置文件

```
[root@exercise1 ~]# vim /etc/ssh/sshd_config   #改32的内容
改：32  SyslogFacility  AUTHPRIV
为：32  SyslogFacility   local0
```

5.2、先重启rsyslog服务(生效配置)

```
[root@exercise1 ~]# systemctl restart rsyslog
```

5.3、再重启sshd服务.生成日志

```
[root@exercise1 ~]# systemctl restart sshd
```

验证是否生成日志并查看其中的内容

[root@exercise1 ~]# cat /var/log/sshd.log

Jan 28 16:29:19 exercise1 sshd[2123]: Received signal 15; terminating.

Jan 28 16:29:19 exercise1 sshd[2148]: Server listening on 0.0.0.0 port 22.

Jan 28 16:29:19 exercise1 sshd[2148]: Server listening on :: port 22.

时间　　主机　服务进程ID　　相关的信息

互动：如何防止日志删除?

```
[root@exercise1 ~]# chattr +a /var/log/sshd.log
[root@exercise1 ~]# lsattr /var/log/sshd.log
-----a---------- /var/log/sshd.log
[root@exercise1 ~]# systemctl restart sshd
[root@exercise1 ~]# cat /var/log/sshd.log   #重启服务，查看日志有所增加
注：这个功能看着很强大，其实不实用，因这样会让系统日志切割时报错，日志有时会大多。
最主的是，黑客可以取消这个属性。
Jan 28 16:29:19 exercise1 sshd[2123]: Received signal 15; terminating.
Jan 28 16:29:19 exercise1 sshd[2148]: Server listening on 0.0.0.0 port 22.
```

```
Jan 28 16:29:19 exercise1 sshd[2148]: Server listening on :: port 22.
Jan 28 17:01:58 exercise1 sshd[2148]: Received signal 15; terminating.
Jan 28 17:01:58 exercise1 sshd[2446]: Server listening on 0.0.0.0 port 22.
Jan 28 17:01:58 exercise1 sshd[2446]: Server listening on :: port 22.
[root@exercise1 ~]# chattr -a /var/log/sshd.log   #取消，这里一定要
#取消，不然后面做日志切割报错
[root@exercise1 ~]# lsattr /var/log/sshd.log
---------------- /var/log/sshd.log
```

互动：当日志增长太多，导致日志文件很大怎么办？

6、日志的切割

在linux下的日志会定期进行滚动增加，我们可以在线对正在进行回滚的日志进行指定大小的切割（动态）。
如果这个日志是静态的。比如没有应用向里面写内容。那么我们也可以用split工具进行切割；

split
语法：split 参数[-nbc] 要切割的文件 输出文件名
参数：

```
-l: 指定文件截断的每一行数，不指定默认为1000行
-a:指定输出文件名的后缀长度（默认为2个：aa,ab..）
-d:后接-数字；切割成几行输出文件，后不接数字，文件名的后缀用数字代替
-b<字节> : 指定每多少字节切成一个小文件
-C:文件大小分割模式（切割时尽量维持每行的完整性）
```

例子：

截断后产生文件名的开头字母，不指定，缺省为x，即截断后产生的文件的文件名为xaa,xab....直到xzz

```
#按行切割
[root@exercise1 ~]# cp /var/log/messages /opt/a.txt
[root@exercise1 ~]# split -6 /opt/a.txt /opt/a
[root@exercise1 ~]# ll /opt/

#按行切割并指定后缀为数字
[root@exercise1 ~]# split -d -4 /opt/a.txt /opt/bak

#按字节切割并指定输出文件名的后缀用数字代替
[root@exercise1 ~]# split -b 20 -d /opt/a.txt /opt/b
[root@exercise1 ~]# ll -h /opt/

#按文件大小切割并指定输出文件名的后缀用数字代替
[root@exercise1 ~]# split -C 1500 -d /opt/a.txt /opt/b
```

如果这个日志是动态的，则使用Logrotate支持按时间和大小来自动切分,以防止日志文件太大。

Logrotate配置文件主要有：

/etc/logrotate.conf以及/etc/logrotate.d/这个子目录下的明细配置文件。

logrotate的执行由crond服务调用的。

```
[root@exercise1 ~]# vim /etc/cron.daily/logrotate   #查看logrotate脚本内容
```

logrotate程序每天由cron在指定的时间（/etc/crontab）启动

日志是很大的,如果让日志无限制的记录下去是一件很可怕的事情，日积月累就有几百兆占用磁盘的空间，

日志切割:

当日志达到某个特定的大小,我们将日志分类,之前的日志保留一个备份,再产生的日志创建一个同名的文件保存新的日志.

拓展：
logrotate是作为linux系统日志的管理工具存在。他可以轮换，压缩，邮件系统日志文件。
默认的logrotate被加入cron的/etc/cron.daily中作为每日任务执行。
/etc/logrotate.conf为其默认配置文件指定每个日志文件的默认规则。
/etc/logrotate.d/* 为/etc/logrotate.conf默认包含目录其中文件也会被logrotate读取。指明每个日志文件的特定规则。

/var/lib/logrotate/statue中默认记录logrotate上次轮换日志文件的时间。

7、实战演示

```
编辑配置文件
[root@exercise1 ~]# vim /etc/logrotate.conf
```

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress
```

说明：(全局参数)

```
weekly：  每周执行回滚，或者说每周执行一次日志回滚
rotate：表示日志切分后历史文件最多保存离现在最近的多少份循环
create：指定新创建的文件的权限与所属主与群组
dateext：使用日期为后缀的回滚文件

#可以去/var/log目录下看看单独配置信息
```

```
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
        minsize 1M
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
```

/var/log/btmp{ 指定的日志文件的名字和路径

missingok 如果文件丢失，将不报错

monthly 每月轮换一次

create 0664 root utmp 设置btmp这个日志文件的权限，属主，属组

minsize 1M 文件超过1M进行回滚，所以大家要知道它不一定每个月都会进行分割，要看这个文件大小来定

rotate1 日志切分后历史文件最多保存1份，不含当前使用的日志

其它参数说明：

monthly:日志文件将按月轮循。其它可用值为‘daily’，‘weekly’或者‘yearly’。

rotate5:一次将存储5个归档日志。对于第六个归档，时间最久的归档将被删除。

compress:在轮循任务完成后，已轮循的归档将使用gzip进行压缩。

delaycompress:总是与compress选项一起用，delaycompress选项指示logrotate不要将最近的归档压缩，压缩将在下一次轮循周期进行。这在你或任何软件仍然需要读取最新归档时很有用。

missingok:在日志轮循期间，任何错误将被忽略，例如“文件无法找到”之类的错误。

notifempty:如果日志文件为空，轮循不会进行。

postrotate/endscript:在所有其它指令完成后，postrotate和endscript里面指定的命令将被执行。在这种情况下，rsyslogd进程将立即再次读取其配置并继续运行。

/var/lib/logrotate/status中默认记录logrotate上次轮换日志文件的时间。

8、实战-使用logrotate进行ssh日志分割

定义了ssh日志存储在/var/log/sshd的基础上执行：

```
注：/etc/logrotate.conf是主配置文件
18 include /etc/logrotate.d     #这一选项标明在调用主配置文件时，会去/etc/logrotate.d查找子配置文件

[root@exercise1 ~]# vim /etc/logrotate.d/sshd    #创建一个sshd配置文件，插入以一下内容
/var/log/sshd.log{    #/var/log/sshd.log有存在才行，不然后面会报错
missingok
daily
create 0600 root root
minsize 1M
rotate 3
}
[root@exercise1 ~]# systemctl restart rsyslog
[root@exercise1 ~]# logrotate  -d /etc/logrotate.d/sshd       #预演，不实际轮循
[root@exercise1 ~]# logrotate  -vf /etc/logrotate.d/sshd      #强制轮循，也就是说即使轮循条件
#没有满足，也可以通过加-f强制让logrotate轮循日志文件
#-v显示指令执行过程
#-f强制执行

[root@exercise1 ~]# ls /var/log/sshd*
/var/log/sshd.log  /var/log/sshd.log.1
```

```
21
22 再次查看日志文件大小，已经为0
23 [root@exercise1 ~]# ll -h /var/log/sshd*
24 -rw-------. 1 root root 0 1月  28 19:07
   /var/log/sshd.log
25 -rw-r--r--. 1 root root 137 1月  28 19:07
   /var/log/sshd.log.1
```

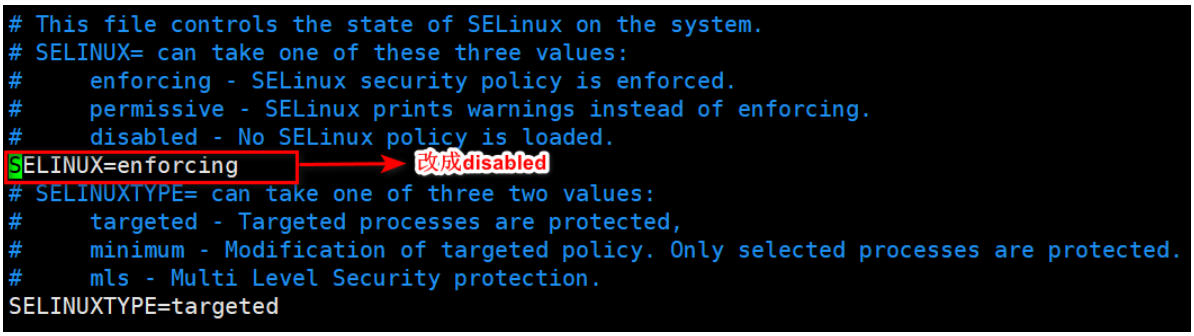
9、配置远程日志服务器-实现日志集中的管理

server端配置

```
1  [root@exercise1 ~]# vim /etc/rsyslog.conf    #使用TCP协议
   方式，收集日志
2  改：19#$ModLoadimtcp
3     20#$InputTCPServerRun514
4  为：
5          $ModLoadimtcp                    (去掉注释)
6          $InputTCPServerRun514       (去掉注释)
7  注：使用UDP协议速度快不保证数据的完整，使用TCP协议可靠完整
8  [root@exercise1 ~]# systemctl restart rsyslog    #重新启
   动rsyslog
9  [root@exercise1 ~]# netstat -anlpt | grep 514    #查看服
   务监听的状态
10 tcp        0      0 0.0.0.0:514             0.0.0.0:*
               LISTEN      1039/rsyslogd
11 tcp6       0      0 :::514                  :::*
                 LISTEN      1039/rsyslogd
```

#在服务端关闭selinux安全策略和防火墙
[root@exercise1 ~]# getenforce
Enforcing
[root@exercise1 ~]# setenforce 0 #关闭selinux功能
[root@exercise1 ~]# getenforce
Permissive

[root@exercise1 ~]# vim /etc/selinux/config #看下图修改




```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
[root@exercise1 ~]# iptables -F   #清空防火墙规则
[root@exercise1 ~]# systemctl stop firewalld   #关闭防火墙
[root@exercise1 ~]# systemctl status firewalld   #查看防火墙活跃状态
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since 五 2022-01-28 19:18:22 CST; 5s ago
     Docs: man:firewalld(1)
  Process: 559 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
 Main PID: 559 (code=exited, status=0/SUCCESS)
```

client端配置

登录exercise2

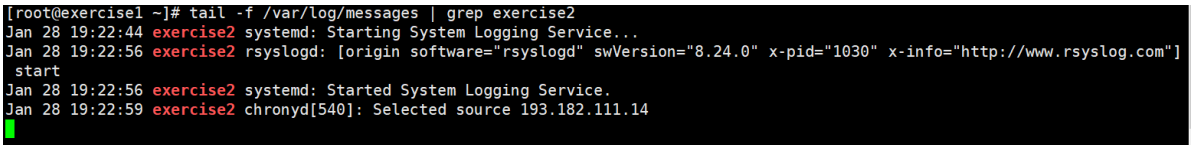
[root@exercise2 ~]# vim /etc/rsyslog.conf #在90行之后，插入

. @@192.168.119.147:514

#注：.所有类别和级别的日志；@@192.168.119.147:514服务端tcp协议的IP和端口

[root@exercise2 ~]# systemctl restart rsyslog 重启rsyslog服务

```
server端查看
查看日志
[root@exercise1 ~]# tail -f /var/log/messages | grep exercise2   #动态查看日志
Jan 28 19:22:44 exercise2 systemd: Starting System Logging Service...
Jan 28 19:22:56 exercise2 rsyslogd: [origin software="rsyslogd" swVersion="8.24.0" x-pid="1030" x-info="http://www.rsyslog.com"] start
Jan 28 19:22:56 exercise2 systemd: Started System Logging Service.
Jan 28 19:22:59 exercise2 chronyd[540]: Selected source 193.182.111.14
```



在client端进行测试

语法：logger　要模拟发送的日志

```
[root@exercise2 ~]# logger "aaaaaa"
```

到server端查看消息

[root@exercise1 ~]# tail -f /var/log/messages | grep exercise2

Jan 28 19:22:44 exercise2 systemd: Starting System Logging Service...

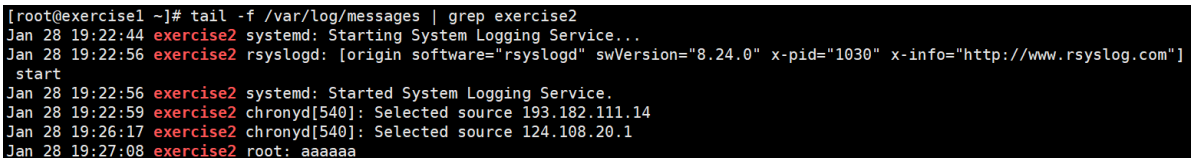
Jan 28 19:22:56 exercise2 rsyslogd: [origin software="rsyslogd" swVersion="8.24.0" x-pid="1030" x-info="http://www.rsyslog.com"] start

Jan 28 19:22:56 exercise2 systemd: Started System Logging Service.

Jan 28 19:22:59 exercise2 chronyd[540]: Selected source 193.182.111.14

Jan 28 19:26:17 exercise2 chronyd[540]: Selected source 124.108.20.1

Jan 28 19:27:08 exercise2 root: aaaaaa



注：
总结：服务器使用udp协议，客户端只能使用的配置文件中这一行只能有一个@，*.*@192.168.1.64:514
服务器使用tcp协议，客户端只能使用的配置文件中这一行必须有两个@@，*.*@@192.168.1.64:514

拓展last与lastb：
在linux系统中，last与lastb命令用来列出目前与过去登录系统的用户相关信息。指令英文原义：

last, lastb - show listing of last logged in users

单独执行last指令时，它会读取位于/var/log/wtmp的文件，并把该给文件的内容记录的登录系统的用户名单全部显示出来。

单独执行lastb指令，它会读取位于/var/log/btmp的文件，并把该文件内容记录的登入系统失败的用户名单，全部显示出来。

更多参数查看请去使用 --help 或者 man

```
[root@exercise1 ~]# last -5     //也可以使用last -n 5
```

显示终端tty1的登录信息
[root@exercise1 ~]# last 1   //等同于last tty1

显示用户weijie和root在8月9号的登录信息
[root@localhost ~]# last -t 20180809090000 weijie //可以看到用户weijie在8.9之前没有登录

wtmp begins Wed Aug 8 18:02:52 2018

[root@localhost ~]# last -t 20180809090000 root //用户root在8.9之前登录过几次

root   pts/0   :0.0   Wed Aug 8 20:19 - down   (00:05)

```
root     tty1         :0               Wed Aug  8 20:19
- down   (00:06)

root     pts/1        :0.0             Wed Aug  8 18:16
- 20:16  (01:59)

root     pts/0        :0.0             Wed Aug  8 18:08
- 20:15  (02:06)

root     tty1         :0               Wed Aug  8 18:04
- 20:16  (02:11)

wtmp begins Wed Aug  8 18:02:52 2018
```