日志管理2

1、日志服务的配置文件

[loring ~]# vim /etc/rsyslog.conf

Provides UDP syslog reception //提供接收远程日志的服务的

#$ModLoad imudp

#$UDPServerRun 514

GLOBAL DIRECTIVES #### //全局配置部分

Include all config files in /etc/rsyslog.d/

$IncludeConfig /etc/rsyslog.d/*.conf //表示包含/etc/rsyslog.d/目录下所有以.conf结尾的配置文件

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

authpriv.* /var/log/secure

mail.* -/var/log/maillog

cron.* /var/log/cron

*.emerg *

uucp,news.crit /var/log/spooler

local7.* /var/log/boot.log

服务或者设备.日志的级别 日志记录到哪个文件

local0~local7：自定义的服务名

2、authpriv.*

服务或者设备.日志的级别

1）服务名称

authpriv(auth)：用户授权相关的，认证等 ssh、login、su等

cron：和计划任务相关的服务

mail：和邮件相关的

news：新闻相关的 uucp

kern：和内核相关的

lpr：打印机相关的

syslog：和rsyslogd进程相关的

local0~local7：自定义服务名称

2）日志的级别：级别越高、日志越少

（1）none：不记录日志

（2）debug：调试信息

（3）info：一般的通知信息

（4）notice：提醒信息，比info稍微重要点

（5）warning（warn）：警告信息，可能有问题

（6）err(error)：错误信息

（7）critical(crit)：比较严重的错误

（8）alert：警报信息，需要立即行动

（9）emerg(panic)：紧急(恐慌)，系统可能已经不可用了

3）服务名称.日志等级的表示

. —— mail.warning 表示记录mail服务，warning及其以上级别的日志

.= —— mail.=warning 表示记录mail服务warning级别日志

.! —— mail.!warning 表示除了warning级别以外的，其他级别的所有日志都记录

.none —— 表示不记录

* —— 表示所有

*. —— 表示所有服务

.* —— 表示所有日志级别

. —— 所有服务的所有级别的日志

小实验

[loring ~]# vim /etc/rsyslog.conf 在第46行添加如下行：

authpriv.* /usr/local/secure //自己规定一个日志文件

[loring ~]# /etc/init.d/rsyslog restart

标签一：

[loring ~]# tail -0f /usr/local/secure

标签二：

[loring ~]# su - test

[test@server150 ~]$ su - root

Password:

su: incorrect password

看标签一的日志变化。

[loring ~]# tail -0f /usr/local/secure

Aug 3 12:41:47 server150 su: pam_unix(su-l:session): session closed for user test

Aug 3 12:41:51 server150 su: pam_unix(su-l:session): session opened for user test by root(uid=0)

Aug 3 12:41:55 server150 su: pam_unix(su-l:auth): authentication failure; logname=root uid=500 euid=0 tty=pts/5 ruser=test rhost= user=root

注意：

1）书写错误

2）selinux必须要关闭的或者是permissive模式，否则无法自动生成你配置的日志文件

为了防止别人修改我的日志，如何做?

[loring ~]# chattr +a /var/log/secure

远程日志 ：可以做日志的备份，也可以做日志的集中管理

集中管理日志的软件：splunk ELK

环境：

172.16.12.250 远程服务器

172.16.254.251 本地服务器

1、本地配置

[loring ~]# vim /etc/rsyslog.conf

. @172.16.12.250 // 此处IP为远程服务器IP

2、远程服务器配置

[loring ~]# vim /etc/rsyslog.conf

Provides UDP syslog reception 开启接收远程日志的功能

$ModLoad imudp

$UDPServerRun 514

3、本地服务器和远程服务器分别重启rsyslog服务

[loring ~]# /etc/init.d/rsyslog restart //254.251

Shutting down system logger: [OK]

Starting system logger: [OK]

[root@srv12 ~]# /etc/init.d/rsyslog restart //12.250

Shutting down system logger: [OK]

Starting system logger: [OK]

4、验证

在本地随意切换几次路径

在远程动态监控日志

注意：

1）服务别忘记重启

2）本地selinux关闭

3）远程主机的防火墙关闭

4）尽量不要互相传日志

日志的轮替、日志的轮滚

为什么要轮滚?

1）防止日志文件过大

2）定期清除日志

日志轮滚的配置文件

```
# vim /etc/logrotate.conf
# grep -v ^# /etc/logrotate.conf | grep -v ^$
    weekly      //轮滚的周期，一周一轮滚，默认每一周执行一次rotate轮转工作
    rotate 4    //保留多少个日志文件(轮转几次).默认保留四个.就是指定日志文件删除之前轮转的次数，0 指没有备份
    create      //旧日志轮滚后是否创建新的空白日志
    dateext     //就是切割后的日志文件以当前日期为格式结尾，如xxx.log-20131216这样,如果注释掉,切割出来是按数字递增,即 xxx.log-1这种格式
    compress    //是否通过gzip压缩转储以后的日志文件，如xxx.log-20131216.gz；如果不需要压缩，注释掉就行
    include /etc/logrotate.d    //包含该路径下的所有配置文件
    /var/log/wtmp {             //仅针对 /var/log/wtmp 所设定的参数
        monthly                 //轮滚周期，一个月
        create 0664 root utmp   //创建新的日志文件  权限664 所有者root 所属组utmp
            minsize 1M          //文件大小超过 1M 后才会切割
        rotate 1
    }
    /var/log/btmp {
```

```
    missingok           //丢了也没关系
    monthly
    create 0600 root utmp
    rotate 1
}
```

其他重要参数说明

---

compress 通过gzip 压缩转储以后的日志

nocompress 不做gzip压缩处理

copytruncate 用于还在打开中的日志文件，把当前日志备份并截断；是先拷贝再清空的方式，拷贝和清空之间有一个时间差，可能会丢失部分日志数据。

nocopytruncate 备份日志文件不过不截断

create mode owner group 轮转时指定创建新文件的属性，如create 0777 nobody nobody

nocreate 不建立新的日志文件

delaycompress 和compress 一起使用时，转储的日志文件到下一次转储时才压缩

nodelaycompress 覆盖 delaycompress 选项，转储同时压缩。

missingok 如果日志丢失，不报错继续滚动下一个日志

errors address 转储时的错误信息发送到指定的Email 地址

ifempty 即使日志文件为空文件也做轮转，这个是logrotate的缺省选项。

notifempty 当日志文件为空时，不进行轮转

mail address 把转储的日志文件发送到指定的E-mail 地址

nomail 转储时不发送日志文件

olddir directory 转储后的日志文件放入指定的目录，必须和当前日志文件在同一个文件系统

noolddir 转储后的日志文件和当前日志文件放在同一个目录下

sharedscripts 运行postrotate脚本，作用是在所有日志都轮转后统一执行一次脚本。如果没有配置这个，那么每个日志轮转后都会执行一次脚本

prerotate 在logrotate转储之前需要执行的指令，例如修改文件的属性等动作；必须独立成行

postrotate 在logrotate转储之后需要执行的指令，例如重新启动 (kill -HUP) 某个服务！必须独立成行

daily 指定转储周期为每天

weekly 指定转储周期为每周

monthly 指定转储周期为每月

rotate count 指定日志文件删除之前转储的次数，0 指没有备份，5 指保留5 个备份

dateext 使用当期日期作为命名格式

dateformat .%s 配合dateext使用，紧跟在下一行出现，定义文件切割后的文件名，必须配合dateext使用，只支持 %Y %m %d %s 这四个参数

size(或minsize) log-size 当日志文件到达指定的大小时才转储，log-size能指定bytes(缺省)及KB (sizek)或MB(sizem).

当日志文件 >= log-size 的时候就转储。 以下为合法格式：

size = 5 或 size 5 （>= 5 个字节就转储）

size = 100k 或 size 100k

size = 100M 或 size 100M

查看某个服务的日志是否轮滚

```
[loring logrotate.d]# cat /etc/logrotate.d/syslog
    /var/log/cron
    /var/log/maillog
    /var/log/messages
    /var/log/secure
    /var/log/spooler
    {
        sharedscripts     //与endscript中间夹的是需要执行的操作
        postrotate              //轮滚后执行的命令
            /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
//重新加载配置
        endscript
    }
```

小实验：

1、修改配置文件，添加一行自己定义的内容，重启服务

```
[loring log]# vim /etc/rsyslog.conf
```

authpriv.* /usr/local/secure

[loring log]# /etc/init.d/rsyslog restart

2、自己编写轮滚配置文件

[loring log]# vim /etc/logrotate.d/secure

```
/usr/local/secure {
    missingok
    notifempty
    daily
    create
    rotate 4
    compress
}
```

3、手动轮滚日志文件

[loring log]# logrotate -vf /etc/logrotate.d/secure

[loring log]# cd /usr/local

[loring local]# ls secure*

secure secure.1.gz

实验续

[loring log]# vim /etc/logrotate.d/ secure

```
/usr/local/secure {
    sharedscripts
        prerotate
            /usr/bin/chattr -a /usr/local/secure   只能追加，不能vi编辑
    endscript
    missingok
    notifempty
    daily
    create
    rotate 4
    compress
    sharedscripts
            postrotate
            /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null  //重读
```

配置

```
        /usr/bin/chattr +a /usr/local/secure
    endscript
}
[loring local]# /etc/init.d/rsyslog restart
[root@server1 local]# chattr +a /usr/local/secure
[loring local]# logrotate -vf /etc/logrotate.d/secure
[loring local]# lsattr secure
-----a-------e- secure
```

/dev/null：空，不管你向它扔什么，都是空

2>：标准错误输出重定向

```
[root@server1 local]# ls secure
secure
[root@server1 local]# ls asdf
ls: cannot access asdf: No such file or directory
[root@server1 local]# ls asdf > /tmp/asdf
ls: cannot access asdf: No such file or directory
[root@server1 local]# ls asdf 2> /tmp/asdf
[root@server1 local]# cat /tmp/asdf
ls: cannot access asdf: No such file or directory
```

2>>：标准错误输出追加重定向