

**1-17-Linux中计划任务与日志的管理**

**1-1 Linux中的计划任务**

* + 1. **at计划任务的使用**
    2. **cron 计划任务的使用**
  1. **Linux服务器的日志管理**
     1. **日志的种类和记录的方式**
     2. **实战-自定义ssh服务的日志存储**
     3. **扩展-日志的回滚**
     4. **实战-搭建rsyslog远程日志服务器**

**1-1 Linux中的计划任务**

**计划任务：在某个时段自动执行某个任务。**

**Linux 系统中;**

**At cron**

**at # 只能执行一次**

**语法：**

**at 时间**

**服务：atd 必须开启**

**[root@xuegod60 ~]# systemctl status atd 查看状态**

**● atd.service - Job spooling tools**

**Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; vendor preset: enabled)**

**Active: active (running) since 三 2016-06-22 19:59:59 CST; 38min ago**

**Main PID: 1339 (atd)**

**CGroup: /system.slice/atd.service**

**└─1339 /usr/sbin/atd -f**

**6月 22 19:59:59 xuegod60.cn systemd[1]: Started Job spooling tools.**

**6月 22 19:59:59 xuegod60.cn systemd[1]: Starting Job spooling tools...**

**[root@xuegod60 ~]# systemctl restart atd 重启**

**[root@xuegod60 ~]# systemctl enable atd 开机自启**

**[root@xuegod60 ~]#**

**实战-使用at创建计划任务**

[**root@xuegod60 ~]# date 查看时间**

**2016年 06月 22日 星期三 20:40:45 CST**

**[root@xuegod60 ~]# at 20:42**

**at> mkdir /root/aa 执行的操作**

**at> <EOT> 结束 ctrl+d**

**job 6 at Wed Jun 22 20:42:00 2016**

**[root@xuegod60 ~]# at -l 查看创建成功的计划任务**

**6 Wed Jun 22 20:42:00 2016 a root**

**任务的编号 执行的日期 root**

**[root@xuegod60 ~]#**

**验证:**

**[root@xuegod60 ~]# date**

**2016年 06月 22日 星期三 20:44:25 CST**

**[root@xuegod60 ~]# ls /root/**

**aa a.txt san 模板 图片 下载 桌面**

**anaconda-ks.cfg initial-setup-ks.cfg 公共 视频 文档 音乐**

**[root@xuegod60 ~]#**

**（在某天的某时执行计划任务）**

**实战1**

**[root@xuegod60 ~]# date**

**2016年 06月 22日 星期三 20:45:27 CST**

**[root@xuegod60 ~]# at 20:50 2016-06-22**

**at> rm -fr /root/aa**

**at> mkdir /tmp/abc**

**at> <EOT>**

**job 7 at Wed Jun 22 20:50:00 2016**

**[root@xuegod60 ~]# at -l**

**7 Wed Jun 22 20:50:00 2016 a root**

**[root@xuegod60 ~]#**

**效果**



**实战2:**

**[root@xuegod60 ~]# date**

**2016年 06月 22日 星期三 20:47:31 CST**

**[root@xuegod60 ~]# at now +10min**

**at> touch /opt/abc.txt**

**at> echo "aaaaa" > /opt/abc.txt**

**at> <EOT>**

**job 8 at Wed Jun 22 20:58:00 2016**

**[root@xuegod60 ~]# at -l**

**7 Wed Jun 22 20:50:00 2016 a root**

**8 Wed Jun 22 20:58:00 2016 a root**

**[root@xuegod60 ~]#**

**对于系统来说 :计划任务　就是在某个特定的时刻　执行一个脚本文件**

**创建成功的at计划任务会在/var/spool/at/目录下成生一个可执行的脚本文件**

**[root@xuegod60 ~]# ls /var/spool/at/\***

**/var/spool/at/a000080174f9ca**

**/var/spool/at/spool:**

**[root@xuegod60 ~]# at -l**

**8 Wed Jun 22 20:58:00 2016 a root**

**[root@xuegod60 ~]# at 20:00**

**at> rm -fr aa**

**at> <EOT>**

**job 9 at Thu Jun 23 20:00:00 2016**

**[root@xuegod60 ~]# at -l**

**8 Wed Jun 22 20:58:00 2016 a root**

**9 Thu Jun 23 20:00:00 2016 a root**

**[root@xuegod60 ~]# ll /var/spool/at/\***

**-rwx------ 1 root root 4288 6月 22 20:49 /var/spool/at/a000080174f9ca**

**-rwx------ 1 root root 4238 6月 22 20:53 /var/spool/at/a000090174ff30**

**/var/spool/at/spool:**

**总用量 0**

**[root@xuegod60 ~]#**

**删除at计划任务：**

**atrm　命令**

**atrm + 计划任务的编号**

**[root@xuegod60 ~]# at -l**

**8 Wed Jun 22 20:58:00 2016 a root**

**9 Thu Jun 23 20:00:00 2016 a root**

**[root@xuegod60 ~]# atrm 8**

**[root@xuegod60 ~]# at -l**

**9 Thu Jun 23 20:00:00 2016 a root**

**[root@xuegod60 ~]# ll /var/spool/at/\***

**-rwx------ 1 root root 4238 6月 22 20:53 /var/spool/at/a000090174ff30**

**/var/spool/at/spool:**

**总用量 0**

**[root@xuegod60 ~]#**

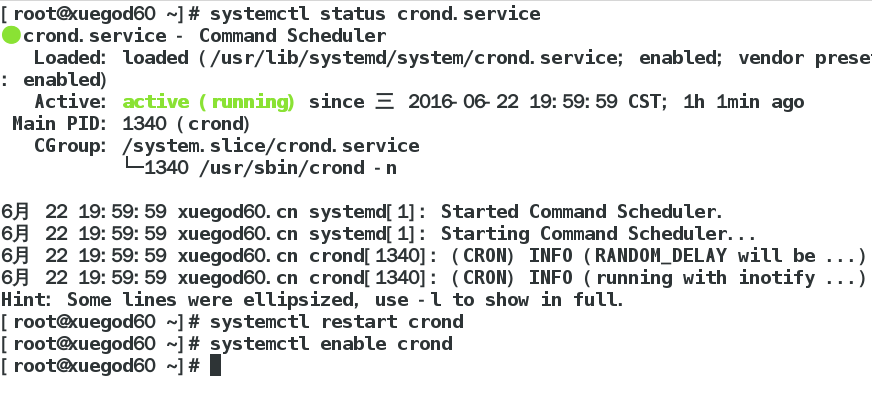
**总结　　：at计划任务执行一次,下次就不能再次执行**

**在生产环境中：　　一般都是周期的去执行某个任务　　　例如：备份日志**

**周期性计划任务：**

**ｃｒｏｎ 主程序**

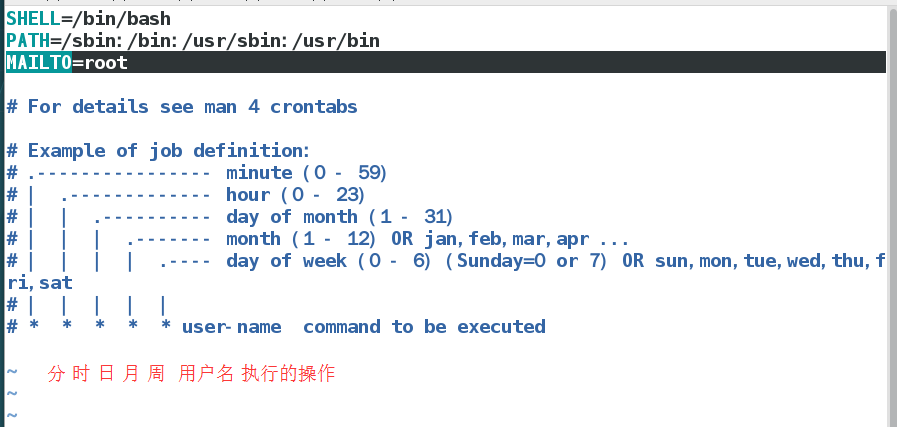
**Crond服务 服务开启**



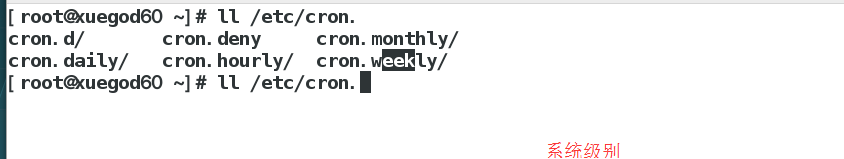
**１.系统级别　２，用户级别的计划任务**

**主配置文件**

**[root@xuegod60 ~]# vim /etc/crontab （内容说明）**



**对于系统级别的计划任务，需要执行的命令和脚本都放在 /etc/cron.\***



**针对用户级别的计划任务：**

**对于root用户：**

**命令：**

**#crontab –e 创建一个计划任务**

**#crontab –l 显示**

**#crontab –r 删除计划任务**

**实战-创建计划任务**

**crontab -e 编辑**

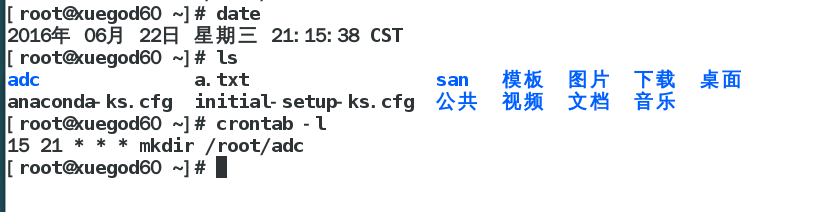
**分 时 日 月 星**

**写入:**

**15 21 \* \* \* mkdir /root/abc  
21:15 创建目录**

**保存退出**

**效果:**



**分 时 日 月 星 谁做后面的事情 命令**

**每个取值范围：**

**分：0－59**

**小时：0－23**

**日：1－31**

**月：1－12**

**周：0－7 0 7 都是周日**

**#每月9,18,22号这几天的凌晨1点1分，执行一个备份脚本**

**1 1 9,18,22 \* \* /root/aa.sh**

**#每月9-22号这几天的凌晨1点1分，执行一个备份脚本**

**1 1 9-22 \* \*　　／root/aa.sh**

**#每5分钟，执行一次**

**\*/5 \* \* \* \* /root/aa.sh**

**\* \*/5 \* \* \* /root/aa.sh**

**anacron**

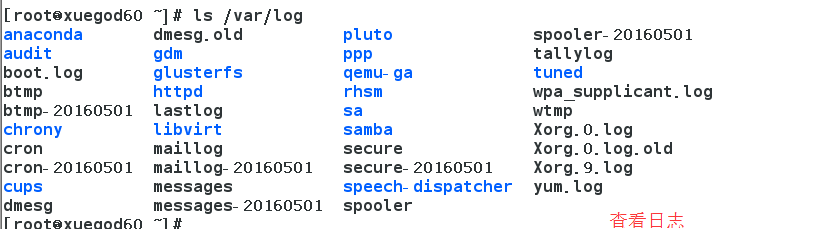
**cron用控制循环执行例行性工作。如果我要设定机器每早8点进行备份用服务。除非我机器保证在8点这个时间点不会关机，如果关机了，cron中的脚本，在下次开机将不会被执行。**

**anacron并没有取代cron的意思，anacron用于，机器重启后，会侦测停机期间，有没有cron没有执行的计划任务，如果有，会立即，执行一下没有执行的任务。**

**日志服务**

1-2-1 日志的管理

/var/log 存放日志的位置.



1. **cron 计划任务的日志文件 vim cron 查看文件的内容**
2. **/var/log/wtmp 系统登录的情况登入登出**

**vim 编辑 是加密的内容**

**last 命令**🡪**查看日志内容**

**3. /var/log/lastlog 最后登录信息**

**lastlog 记录所有的用户什么时候登录过系统**

**lastlog 和 last的区别**

**last 查看是黑客IP**

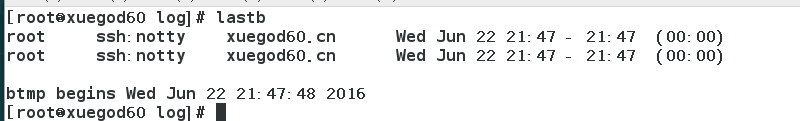
**lastlog 查看黑客的后门的账号**

**/var/log/btmp 用户登录系统的错误信息**

**模拟使用ssh错误的连接本机**

**ssh root@主机IP**

**输错密码**



**防止暴力破解**

**可使用 ll -h 查看文件的大小**

**1-2-2 日志的记录方式:**

**分类🡪 级别🡪**

**日志的种类:**

**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**

**local 0🡪local 7 8个系统保留的类 供其它的程序使用 /或者是用户自定义**

**日志的级别: 轻🡪重**

**debug 排错的信息**

**info 正常的信息**

**notice 注意**

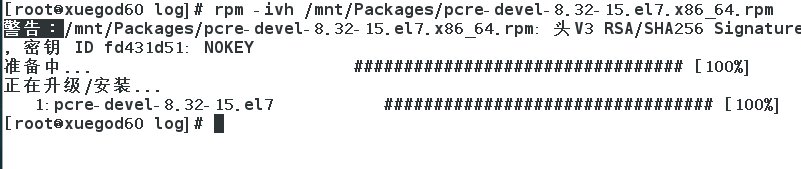
**warn 警告**

**err 错误**

**crit 关键的错误**

**alert 警报**

**energ 紧急**



**日志服务**

**旧版本 rhel5 服务名称 syslog**

**配置文件 /etc/syslog.conf**

**rhel6-7**

**服务名称 rsyslog**

**配置文件 /etc/rsyslog.conf**

**编辑配置文件 vim /etc/rsyslog.conf**

**类型.级别**

**#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**

**.none 排除**

**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**

**-号 邮件的信息比较多,现将数据存储到内存,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**

**日志输入的规则**

**. info 高于info级别的信息全部记录到某个文件**

**.= info 级别 仅记录等于某个级别的日志**

**例:.=info 只记录info级别的日志**

**.! info 除了某个级别以外,记录所有的级别信息**

**例.!err 除了err外 记录所有**

**cron.none 指的是排除某个级别**

**1-2-3 自定义ssh服务的日志**

**编辑文件 vim /etc/rsyslog.conf**

**输入**

**local 0 .\* /var/log/sshd.log 日志的保存路径**

**定义ssh服务的日志级别**

1. **编辑sshd服务的主配置文件vim /etc/ssh/sshd\_config**
2. **/定位查找 关键字Syslog**



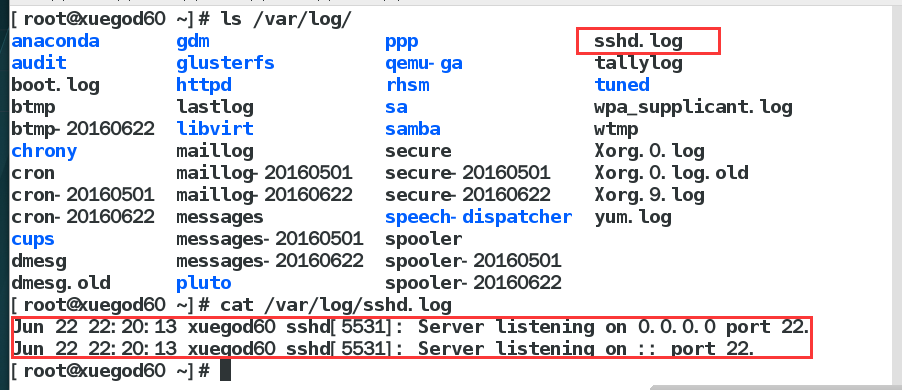
**3. 更改类型是local 0 ,保存退出**

**SyslogFacility local0**

1. **重启 rsyslog sshd服务**

**systemctl restart rsyslog.service**

**systemctl restart sshd**



**日志的格式: 时间 主机 服务 进程ID 相关的信息**

**如何防止日志被黑客删除**

**chattr +a /var/log/sshd.log**

**lsattr /var/log/sshd.log**

**日志的回滚**

**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**

**如果你要找出某一条可用信息🡪海底捞针**

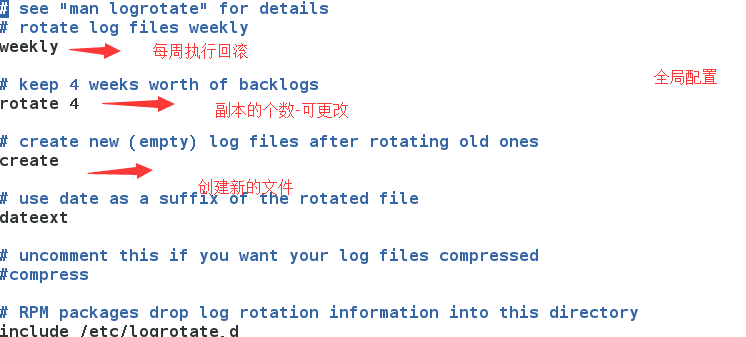
**日志回滚原理:当日志达到某个特定的大小,我们将日志分类,之前的日志保留一个备份,再产生的日志创建一个同名的文件保存新的日志.**

**100M🡪14nian 🡪放到a.txt**

**之后产生的日志🡪b.txt**

**实战🡪**

1. **编辑配置文件/etc/logrotate.conf**



**weekly : 每周执行回滚**

**rotate : 保留4个副本**

**create : 创建新的文件存储数据**

1. **单独配置**

**/var/log/wtmp { 日志文件**

**monthly 每 个月**

**create 0664 root utmp 创建一个文件**

**minsize 1M 文件超过1M🡪进行回滚**

**rotate 1 回滚产生的副本**

**}**

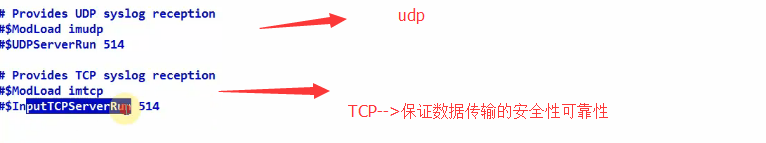
**配置远程日志服务器->实现日志集中的管理**

**server xuegod60**

**client xuegod61**

**server端配置**

**1,编辑 /etc/rsyslog.conf**



**UDP🡪速度快🡪不保证数据的完整**

**TCP🡪可靠.完整**

**使用TCP的方式**

**去除#号注释**

1. **重新启动 rsyslog**

**systemctl restart rsyslog**

1. **查看服务监听的状态**

**netstat -anlpt| grep 514**



**client端配置:**

**vim/etc/rsyslog.conf**

**添加**

**\*.\* 所有类别和级别的日志**

**@@192.168.1.60:514 服务端的IP:端口**

**重启rsyslog 服务**

**systemctl restart rsyslog**

**服务端验证:**

**关闭防火墙 selinux**

**tail -f /var/log/message 动态的**

**client 重启服务 随便**