

Chapter 03. 사용자 모니터링

# 사용자 모니터링과 보안, 그리고 시스템 로깅

## 사용자 모니터링

### 사용자 모니터링 및 시스템 모니터링

- 접속 현황
  - users
    - tty, pts
  - who
  - w
- 접속 로그
  - last
  - lastb
- 터미널 로그
  - history
- 로그 파일
  - 인증로그
    - /var/log/auth.log
  - 시스템 로그 파일
    - /var/log/\*
  - 응용 소프트웨어 로그 파일
    - /var/log/[데몬명]/\*
- 로그 관리 유틸리티
  - logrotate
- 스케줄 작업 관리
  - cron, anacron

## 사용자 도구 - 모니터링

접속 모니터링 도구 - users, who, w

- 현재 접속한 사용자 목록

- users

```
user1@user1-VirtualBox:~$ users
user1 user2
```

- 현재 접속한 사용자 목록과, 터미널 번호, 접속 시간 및 접속 장소

- who

```
user1@user1-VirtualBox:~$ who
user1      :0                2020-06-03 23:11 (:0)
user2      pts/0          2020-06-05 21:37 (10.0.2.2)
```

- 시스템 정보, 현재 접속한 사용자 목록과, 터미널 번호, 접속 장소, 접속 시간, 자원 소모량 및 하는 행위

- w

```
user1@user1-VirtualBox:~$ w
 21:42:31 up 1 day, 22:33,  2 users,  load average: 0.09, 0.07, 0.02
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
user1      :0        :0               수23    ?xdm?  10:13   0.01s /usr/lib/gdm3/gdm-x-
user2      pts/0     10.0.2.2         21:37    3.00s  0.05s  0.05s -bash
```

# 사용자 도구 - 모니터링

## 접속 모니터링 도구 - who 상세 옵션

- 현재 접속한 사용자와 시스템의 상세 정보
  - who
    - -a : all (-b, -d, --login, -p, -r, -t -T, -u)
    - -b : 최근 부팅 시간
    - -r : 런레벨
    - -H : 헤더 정보 표시
- 현재 터미널 확인 명령어
  - tty
- 접속 터미널 용어 설명
  - tty = teletypewriter = terminal = 텍스트 입/출력 환경
    - tty0, tty1, tty2, ... => 터미널 콘솔
    - ttyS0, ttyS1, ... => 시리얼 콘솔
  - pts = pseudo terminal slave = xterm, screen, ssh 등의 터미널 에뮬레이션 인터페이스
    - pts/0, pts/2, ... => 가상 (원격접속) 터미널

```
user1@user1-VirtualBox:~$ who -b
system boot 2020-06-03 23:09
user1@user1-VirtualBox:~$ who -r
실행-수준 5 2020-06-03 23:10
user1@user1-VirtualBox:~$ who -H -a
이름      행      시간      IDLE      PID 주석      나가기
system boot 2020-06-03 23:09
실행-수준 5 2020-06-03 23:10
user1      ? :0      2020-06-03 23:11      ?      8958 (:0)
user2      + pts/0      2020-06-05 21:37 00:08      17145 (10.0.2.2)
```

```
(base) hyunlove@instance-1:~$ who -H -a
NAME      LINE      TIME      IDLE      PID COMMENT      EXIT
system boot 2020-06-04 04:16
LOGIN     tty1      2020-06-04 04:16      1053 id=tty1
LOGIN     ttyS0     2020-06-04 04:16      1044 id=tyS0
run-level 5 2020-06-04 04:17
hyunlove + pts/0 2020-06-05 13:15      .      11814 (74.125.41.164)
```

## 사용자 도구 - 모니터링

### 사용자간 메시지 - wall (write all)

- 시스템 관리용 메시지
  - wall "System will shut down in 5 minutes."

```
user1@user1-VirtualBox:~$ wall "system will shut down in 5 mins"
user1@user1-VirtualBox:~$
```

```
Broadcast message from user1@user1-VirtualBox (pts/1) (Fri Jun  5 22:02:51 2020)
system will shut down in 5 mins
```

- 개별 사용자간 메시지 전달 (채팅)
  - write [username] [terminal]

## 사용자 도구 - 모니터링

사용자 접속 로그 - last (최근 로그인 시간)

- last
  - /var/log/wtmp (바이너리 로그 파일)

```
user1@user1-VirtualBox:~$ last
user1    tty4                Fri Jun  5 21:58    still logged in
user2    pts/2                10.0.2.2           Fri Jun  5 21:57 - 23:50 (01:52)
user2    pts/0                10.0.2.2           Fri Jun  5 21:37    still logged in
user1    :0                   :0                 Wed Jun  3 23:11    still logged in
reboot   system boot          4.15.0-101-gener   Wed Jun  3 23:09    still running

wtmp begins Wed Jun  3 02:20:26 2020
user1@user1-VirtualBox:~$
```

- last [username]

```
user1@user1-VirtualBox:~$ last user1
user1    tty4                Fri Jun  5 21:58    still logged in
user1    :0                   :0                 Wed Jun  3 23:11    still logged in

wtmp begins Wed Jun  3 02:20:26 2020
```

- last -n [number]

## 사용자 도구 - 모니터링

### 사용자 접속 로그 - lastb (최근 로그인 실패 시간)

- sudo lastb (관리자 권한 필요)
  - /var/log/btmp (바이너리 로그 파일)

```
user1@user1-VirtualBox:~$ sudo lastb
[sudo] user1의 암호:
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:37 - 21:37
user2      ssh:notty    10.0.2.2      Fri Jun  5 21:36 - 21:36

btmp begins Fri Jun  5 21:36:57 2020
user1@user1-VirtualBox:~$
user1@user1-VirtualBox:~$ ls -al /var/log/?tmp
-rw-rw---- 1 root utmp 2688  6월  5 21:37 /var/log/btmp
-rw-rw-r-- 1 root utmp 6528  6월  5 23:50 /var/log/wtmp
```

- lastb [username]
- lastb -n [number]

```
(base) hyunlove@instance-1:~$ sudo lastb
pi      ssh:notty    175.10.162.75  Fri Jun  5 11:43 - 11:43 (00:00)
pi      ssh:notty    175.10.162.75  Fri Jun  5 11:43 - 11:43 (00:00)
administ ssh:notty    37.49.226.157  Fri Jun  5 11:10 - 11:10 (00:00)
admin   ssh:notty    37.49.226.157  Fri Jun  5 11:10 - 11:10 (00:00)
12345   ssh:notty    37.49.226.173  Fri Jun  5 10:07 - 10:07 (00:00)
1234    ssh:notty    37.49.226.173  Fri Jun  5 10:07 - 10:07 (00:00)
123     ssh:notty    37.49.226.173  Fri Jun  5 10:06 - 10:06 (00:00)
pasmak@w ssh:notty    37.49.226.173  Fri Jun  5 10:06 - 10:06 (00:00)
pi      ssh:notty    191.23.168.249 Fri Jun  5 09:09 - 09:09 (00:00)
pi      ssh:notty    191.23.168.249 Fri Jun  5 09:09 - 09:09 (00:00)
administ ssh:notty    37.49.226.157  Fri Jun  5 07:15 - 07:15 (00:00)
admin   ssh:notty    37.49.226.157  Fri Jun  5 07:15 - 07:15 (00:00)
admin   ssh:notty    194.180.224.130 Fri Jun  5 03:42 - 03:42 (00:00)
        ssh:notty    65.49.20.67    Fri Jun  5 03:04 - 03:04 (00:00)
telnet  ssh:notty    37.49.226.173  Thu Jun  4 22:45 - 22:45 (00:00)
svn     ssh:notty    37.49.226.173  Thu Jun  4 22:45 - 22:45 (00:00)
support ssh:notty    37.49.226.173  Thu Jun  4 22:45 - 22:45 (00:00)
student ssh:notty    37.49.226.173  Thu Jun  4 22:44 - 22:44 (00:00)
oracle  ssh:notty    37.49.226.163  Thu Jun  4 22:33 - 22:33 (00:00)
admin   ssh:notty    37.49.226.163  Thu Jun  4 22:32 - 22:32 (00:00)
admin   ssh:notty    194.180.224.130 Thu Jun  4 22:31 - 22:31 (00:00)
12345   ssh:notty    37.49.226.173  Thu Jun  4 22:30 - 22:30 (00:00)
1234    ssh:notty    37.49.226.173  Thu Jun  4 22:30 - 22:30 (00:00)
123     ssh:notty    37.49.226.173  Thu Jun  4 22:29 - 22:29 (00:00)
pasmak@w ssh:notty    37.49.226.173  Thu Jun  4 22:29 - 22:29 (00:00)
admin   ssh:notty    37.49.226.129  Thu Jun  4 22:27 - 22:27 (00:00)
admin   ssh:notty    41.232.104.26  Thu Jun  4 17:52 - 17:52 (00:00)
admin   ssh:notty    41.218.216.39  Thu Jun  4 17:52 - 17:52 (00:00)
admin   ssh:notty    194.180.224.130 Thu Jun  4 13:00 - 13:00 (00:00)
centos  ssh:notty    93.157.62.102  Thu Jun  4 12:05 - 12:05 (00:00)
ansible ssh:notty    93.157.62.102  Thu Jun  4 12:05 - 12:05 (00:00)
admin   ssh:notty    93.157.62.102  Thu Jun  4 12:05 - 12:05 (00:00)
admin   ssh:notty    93.157.62.102  Thu Jun  4 12:05 - 12:05 (00:00)
centos  ssh:notty    93.157.62.102  Thu Jun  4 11:42 - 11:42 (00:00)
ansible ssh:notty    93.157.62.102  Thu Jun  4 11:41 - 11:41 (00:00)
admin   ssh:notty    93.157.62.102  Thu Jun  4 11:41 - 11:41 (00:00)
admin   ssh:notty    93.157.62.102  Thu Jun  4 11:41 - 11:41 (00:00)

btmp begins Thu Jun  4 11:41:30 2020
```

## 사용자 로그 - 터미널 로그

사용자의 명령어, history

- 사용자의 명령어 추적
  - history
  - .bash\_history 파일

```
user1@user1-VirtualBox:~$ history
 1  sudo apt update
 2  sudo apt list --upgradable
 3  sudo apt upgrade
 4  ps x
 5  sudo apt upgrade
 6  sudo apt update
 7  apt list --upgradable
 8  sudo apt upgrade
 9  ls -al /var/lib/dpkg/lock-frontend
10  lsof -an
11  sudo lsof -a
```

```
user1@user1-VirtualBox:~$ cat .bash_history
sudo apt update
sudo apt list --upgradable
sudo apt upgrade
ps x
sudo apt upgrade
sudo apt update
apt list --upgradable
sudo apt upgrade
ls -al /var/lib/dpkg/lock-frontend
lsof -an
sudo lsof -a
```



## 사용자 로그 - 권한 로그

인증 로그를 통한 감사 추적 - login, sudo 등 인증 (권한상승) 을 요청하는 명령어의 기록

- 인증 로그 확인
  - /var/log/auth.log 파일

```
user1@user1-VirtualBox:~$ tail -F /var/log/auth.log
Jun  5 22:09:01 user1-VirtualBox CRON[18054]: pam_unix(cron:session): session closed
for user root
Jun  5 22:11:53 user1-VirtualBox sudo:    user1 : TTY=pts/1 ; PWD=/home/user1 ; USER
=root ; COMMAND=/usr/bin/lastb
Jun  5 22:11:53 user1-VirtualBox sudo: pam_unix(sudo:session): session opened for us
er root by (uid=0)
Jun  5 22:11:53 user1-VirtualBox sudo: pam_unix(sudo:session): session closed for us
er root
Jun  5 22:15:01 user1-VirtualBox CRON[18201]: pam_unix(cron:session): session opened
for user root by (uid=0)
Jun  5 22:15:01 user1-VirtualBox CRON[18201]: pam_unix(cron:session): session closed
for user root
```

## 시스템 로그

다양한 시스템 로그 살펴보기 - /var/log/\*

- bootstrap.log - 부팅 로그 (시스템 부팅 과정에서 발생하는 성공/실패 로그)
- dpkg.log - 패키지 설치 로그
- kern.log - 커널 로그
  - 현재 부팅 후 커널 로그는 커맨드라인 dmesg 로 확인 (시스템 디바이스 메시지 등)
- syslog - 애플리케이션 로그 (각종 시스템 소프트웨어, 응용 소프트웨어 의 로그)
- Xorg.0.log - X윈도우 각종 로그 (윈도우 애플리케이션의 오류 등)

## 시스템 로그 - 응용 소프트웨어 로그

다양한 응용 소프트웨어 로그 살펴보기 - /var/log/(애플리케이션)/\*

- apt/history.log
  - 업그레이드 등에 수행된 명령어 로그 기록
- apt/term.log
  - 위 수행된 결과의 로그 기록
- nginx/access.log
  - 접속 로그, GET / POST, 요청 URL, 응답값 (허용 200, 실패 404 등), 등
- nginx/error.log
  - 서버 시스템의 (치명적) 오류
- apache2/access.log
  - 접속 로그 (상동)
- apache2/error.log
  - 서버 시스템의 (치명적) 오류

## 로그 유틸리티

### 시스템 로그의 자동 (용량) 관리 - logrotated

- 시스템 로그 설정파일
  - /etc/logrotate.conf
- 애플리케이션별 로그 관리 옵션
  - /etc/logrotate.d/\*

```
user1@user1-VirtualBox:~$ cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
```

```
user1@user1-VirtualBox:~$ ls /etc/logrotate.d/
alternatives  dpkg          pm-utils      ufw
apache2       lightdm       postgresql-common  unattended-upgrades
appport       mysql-server  ppp           upstart
apt           nginx         rsyslog       vsftpd
cups-daemon   php7.2-fpm    speech-dispatcher
```

```
user1@user1-VirtualBox:~$ cat /etc/logrotate.d/rsyslog
/var/log/syslog
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscript
}
```

```
/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
```

```
user1@user1-VirtualBox:~$ cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 www-data adm
    sharedscripts
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi \
    endscript
    postrotate
        invoke-rc.d nginx rotate >/dev/null 2>&1
    endscript
}
```

## 스케줄 프로세스

특정 시간마다 특정 프로세스 실행 - cron, anacron

- cron 이란?
  - cron = Command Run On
  - 특정 시간마다 반복하여 작업을 수행, 특정 월, 요일, 시간, 등의 조건 설정
  - 단, 해당 일자에 시스템이 꺼져 있으면 동작하지 않음
  - 시작 위치
    - /etc/crontab
- anacron 이란?
  - 해당 작업이 정해진 시간 내에 실행된 적이 있는지를 확인하여 없다면 적절한 시점(정해진 시간 후)에 실행
  - 시작 위치
    - /etc/anacrontab

# 스케줄 프로세스 - cron

특정 시간마다 특정 스크립트 실행 - cron 상세기능

- 실행 데몬
  - cron (systemctl status cron)
- 시스템 작업
  - /etc/crontab
    - /etc/cron.hourly
    - /etc/cron.daily
    - /etc/cron.weekly
    - /etc/cron.monthly

- 동작 조건

분	시	일	월	주	권한	명령어
17	*	*	*	*	root	xxxxx (매시 17분에 xxxxx 수행)
25	6	*	*	*	root	yyyyy (매일 6:25분에 yyyyy 수행)
47	6	*	*	7	root	zzzzz (매주 일요일 6:47분에 zzzzz 수행) -> 요일 0(일요일) ~ 6(토요일), 7 = 0
52	6	1	*	*	root	qqqqq (매달 1일, 6:52분에 qqqqq 수행)

# 스케줄 프로세스 - anacron

특정 시간마다 특정 스크립트 실행 - anacron 상세기능

- 실행 데몬
  - 없음 (cron 를 통해 실행)
- 시스템 작업
  - /etc/anacrontab
- 동작 조건

일	분	체크디렉토리	명령어
1	5	cron.daily	xxxxx
7	10	cron.weekly	yyyyy
...			

-> 최근 1일동안 cron.daily 가 실행되지 않았다면, 5분 후 xxxxx 실행  
-> 최근 7일동안 cron.weekly 가 실행되지 않았다면, 10분 후 yyyyy 실행

## 스케줄 프로세스 - 사용자별 cron

특정 시간마다 특정 프로세스 실행 - cron 상세기능

- 실행 데몬
  - crond
- 사용자별 작업
  - 스케줄 작업 만들기
    - crontab -e
  - 스케줄 작업 확인
    - crontab -l
  - 스케줄 작업 삭제
    - crontab -r
  - 저장 공간
    - /var/spool/cron/crontabs/[사용자명]

```
user1@user1-VirtualBox:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
```