

오늘도 난, 하하하

- [Home](#)
- [Admin](#)
- [Write](#)

[UNIX / Linux] 특수 권한(setuid, setgid, sticky bit)

[보안/- System](#) 2015.06.27 00:18



프로세스 번호

- UNIX 시스템에서는 프로세스에 다섯 가지 번호 부여

1. 프로세스에 부여되는 번호들

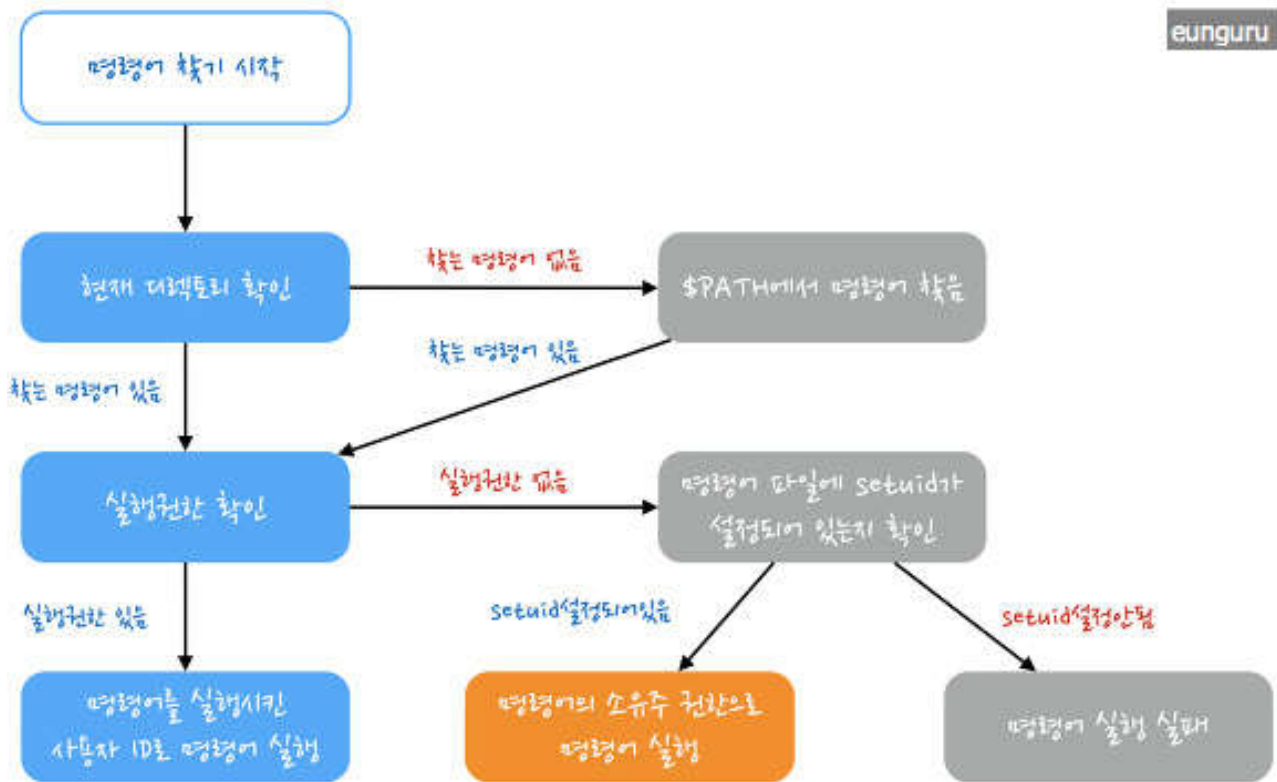
- 1) 프로세스 식별자(PID)
- 2) 실제 사용자 ID(RUID)
- 3) 유효 사용자 ID(EUID)
- 4) 실제 사용자 그룹(RGID)
- 5) 유효 사용자 그룹 ID(EGID)

2. 사용 용도

- 계정 관리에 사용: RUID, RGID
- 접근 권한 결정에 사용: EUID, EGID (보안에 주의)

- 일반적으로 실제 번호와 유효 번호는 동일함

시스템에서 사용자가 명령 실행 시 명령어를 찾는 경로와 절차



접근 권한

- 권한 관리 명령 chmod, umask, 소유주 변경 명령 chown, chgrp (이전 글 참조)

[2015/04/30 - \[보안/- System\] - \[UNIX / Linux\] 권한 관리\(chmod, chown, chgrp, umask\)](#)

- 접근 권한은 8진수 또는 r(읽기권한, 4), w(쓰기권한, 2), x(실행권한, 1) 문자로 표현 가능
- 8진수로 표현할 때는 권한의 합으로 표시함(예- 읽기+쓰기권한 6, 읽기+실행 권한 5 등)
- 8진수 3자리(3bit)로 소유자, 그룹 소유자, 기타 사용자를 위한 파일 모드를 설정

1. 접근 권한 설정 예제

- 접근 권한을 755로 표현하는 것과 0755로 표현하는 것은 동일한 표현, 네 자리가 되지 않는 표현은 앞에 0이 생략됨

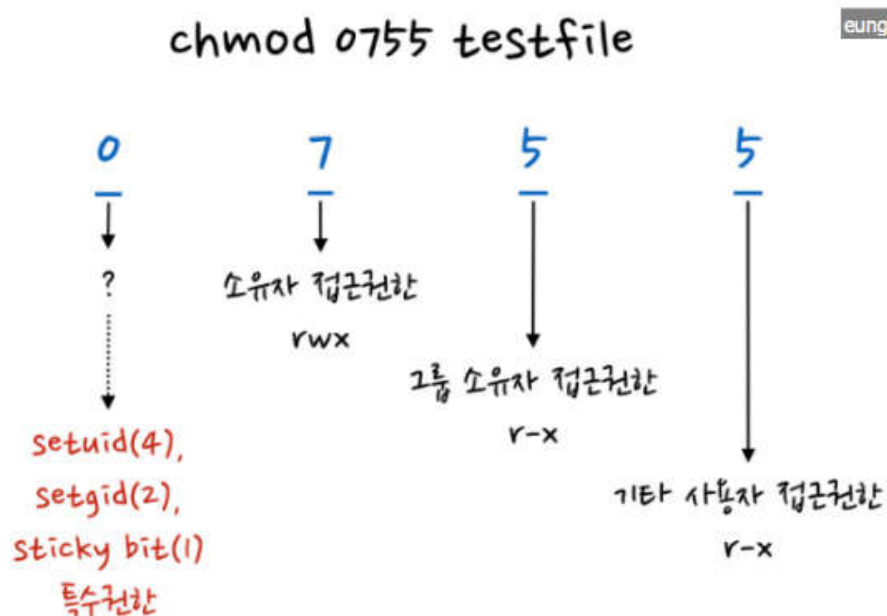
```
eunguru — root@kali: ~ — ssh — 80x36 eunguru
root@kali:~# umask
0022
root@kali:~# touch testfile
root@kali:~# ls -l testfile
-rw-r--r-- 1 root root 0  6월 20 21:28 testfile
root@kali:~# chmod 744 testfile
root@kali:~# ls -l testfile
-rwxr--r-- 1 root root 0  6월 20 21:28 testfile
root@kali:~# chmod 0755 testfile
root@kali:~# ls -l testfile
-rwxr-xr-x 1 root root 0  6월 20 21:28 testfile
root@kali:~#
```

특수권한

1. 특수권한

- UNIX 시스템은 파일에 대한 접근 권한 및 파일 종류를 나타내기 위해 16bit를 사용한다.
- 각 3bit씩 총 9bit는 소유자 접근권한(user), 그룹 소유자 접근권한(group), 기타 사용자 접근권한(other)의 권한을 기술하는데 사용
- 4bit는 파일의 종류 표현에 사용
- 3bit는 특수권한에 사용

2. 각 비트에 대한 설명



파일종류	특수권한			소유자 접근권한			그룹 소유자 접근 권한			기타 사용자 접근 권한		
	4	2	1	4	2	1	4	2	1	4	2	1
-,d,c,b,s,l,p	setuid	setgid	sticky bit	r	w	x	r	w	x	r	w	x

setuid 비트

- setuid 비트: 8진수 4000

- setuid 비트를 실행 파일에 적용하면 실행 사용자(프로그램을 실제 실행 중인 사용자)에서 프로그램 소유자의 ID로 유효사용자(EUID)가 변경됨

1. setuid 비트를 설정하여 사용하는 경우

- 슈퍼유저 root만 접근할 수 있는 파일이나 명령에 대해, 일반 사용자로 접근하는 것이 기능상 필요한 경우

(setuid 비트가 설정된 파일은 실행순간만 그 파일의 소유자 권한으로 실행,

실행 순간만 권한을 빌려온다라고 이해하면 쉬움)

- 매번 슈퍼유저 root가 어떤 행위를 해주지 않아도 되고, 일반 사용자에게 root권한을 주지 않아도 되기때문에 setuid 비트를 적용하는 것이 시스템 운영면에서 효율적

- 대부분 슈퍼유저가 소유한 소수 프로그램들에만 주어짐, 일반 사용자가 그 프로그램을 실행하면 setuid root가 되고,

슈퍼유저의 유효한 특권들을 가지고 실행되기때문에 일반 사용자의 접근이 금지된 파일과 디렉토리들에 접근 가능하게끔

해줌

2. setuid 비트 설정 시 보안 취약점

- root권한이 필요없는 프로그램에 소유주가 root로 되어 있고 setuid가 설정된 경우는 보안상으로 매우 취약

- 일반 사용자로 접근하는 경우도 setuid 설정으로 실행 가능해지기 때문이다.

- 권한 상승 우려때문에 setuid 프로그램의 수는 반드시 최소화해야 함

3. setuid 비트 설정 방법

- 8진수(4000)나 기호(u+s)를 이용하여 setuid 비트를 설정할 수 있음(setuid 비트 설정 제거 u-s)

- 권한 변경을 위해 chmod 명령어를 이용함

- setuid 비트가 설정되어 있으면 사용자 접근권한의 실행 권한 자리에 실행 권한이 있으면 소문자 s로 실행 권한이 없으면 대문자 S로 표시됨

```
eunguru — root@kali: ~/dir — seunguru
root@kali:~/dir# umask
0022
root@kali:~/dir# touch setuid1
root@kali:~/dir# touch setuid2
root@kali:~/dir# ls -lF
합계 0
-rw-r--r-- 1 root root 0  6월 26 11:29 setuid1
-rw-r--r-- 1 root root 0  6월 26 11:29 setuid2
root@kali:~/dir# chmod 4744 setuid1
root@kali:~/dir# chmod u+s setuid2
root@kali:~/dir# ls -lF
합계 0
-rwsr--r-- 1 root root 0  6월 26 11:29 setuid1*
-rwsr--r-- 1 root root 0  6월 26 11:29 setuid2
root@kali:~/dir#
```

4. setuid 비트 설정의 활용

1) 패스워드 설정, 변경시 사용

- 패스워드 지정, 변경에 사용하는 /usr/bin/passwd 명령의 경우 setuid 비트가 설정 되어 있음(접근권한: 4755)

- passwd 명령어(파일)로 패스워드 지정, 변경 시 /etc/passwd, /etc/shadow파일이 변경됨

- /etc/passwd의 접근권한: 0644 (슈퍼유저 root만 수정 가능)

- /etc/shadow의 접근권한: 0640 (슈퍼유저 root만 수정 가능)

- passwd 명령어(파일)은 setuid 비트가 설정되어 있으므로 실행 시 소유자인 root의 권한으로 실행됨

- 일반 사용자 계정에서 passwd 명령어 실행 시에 소유자 root 권한으로 실행되기때문에 슈퍼유저 root만 수정가능한 /etc/passwd, /etc/shadow 파일의 수정이 가능해 짐


```
Macintosh HD — root@kali: ~ — ssh — 80x40 eunguru
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~# useradd test
root@kali:~# passwd test
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
root@kali:~#
root@kali:~# ls -lF /usr/bin/passwd
-rwxr-xr-x 1 root root 51096 5월 26 2012 /usr/bin/passwd*
root@kali:~# ls -lF /etc/passwd
-rw-r--r-- 1 root root 2270 6월 22 23:20 /etc/passwd
root@kali:~# ls -lF /etc/shadow
-rw-r----- 1 root shadow 1598 6월 22 23:20 /etc/shadow
root@kali:~# tail -1 /etc/passwd
test:x:1001:1002::/home/test:/bin/sh
root@kali:~# tail -1 /etc/shadow
test:$6$R9E6oja7$GHyF64XlddhmXyRz7UyStT3vDasM6hG3D2Jiy0nrXDo5SswSnelhw8fQu.FeDcr
b8XJJ329W9STW506Qe7buD1:16608:0:99999:7:::
root@kali:~#
root@kali:~# su - test
디렉터리 없음, 루트 디렉터리 (/)로 로그인합니다
$ passwd
test에 대한 암호 변경 중
(현재) UNIX 암호:
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
$ ls -lF /etc/passwd
-rw-r--r-- 1 root root 2270 6월 22 23:20 /etc/passwd
$ ls -lF /etc/shadow
-rw-r----- 1 root shadow 1598 6월 22 23:21 /etc/shadow
$ tail -1 /etc/shadow
tail: cannot open '/etc/shadow' for reading: 허가 거부
$ su - root
암호:
root@kali:~# tail -1 /etc/shadow
test:$6$R3y5t450$wL/gfmh0Grzp0D.Zu03CGncqG0EzcaX9P04a3gF058JFIDCzyzSUwxLIcTiZ1zH
z8aL6JZU5Sneel.LMhQM17.:16608:0:99999:7:::
root@kali:~#
```

슈퍼 유저 root가 아닌
일반 사용자 계정 생성,
패스워드 지정
소유자 root setuid 설정 되어 있음
접근 권한: 4755

접근 권한: 0644
접근 권한: 0640

root에서 test계정으로 사용자 변경
일반 사용자 계정에서 passwd
명령 실행, 패스워드 변경

패스워드 변경으로 파일 변경됨
권한이 없어서
파일 읽기 불가

2) 일반 사용자가 읽을 수 없는 파일 읽기

- 설명을 위해 예제를 만든 것이기 때문에 조금 억지스러운 부분이 있을 수 있음

- /bin/cat 파일에 직접 setuid비트를 설정하지 않고 심볼릭 링크를 만들어 심볼릭링크 파일에 setuid 비트 설정,

해제 시에도 심볼릭링크 파일에 해제

```
eunguru — root@kali: /tmp/test — ssh — 80x42 eunguru
root@kali:/tmp/test# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:/tmp/test# ls -alF
합계 12
drwxrwsrwt 2 root root 4096 6월 26 20:08 ./
drwxrwxrwt 9 root root 4096 6월 26 20:12 ../
-r--r----- 1 root root 19 6월 26 20:03 readonly
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
root@kali:/tmp/test# ls -l /bin/cat
-rwxrwxrwx 1 root root 51856 1월 27 2013 /bin/cat
root@kali:/tmp/test# ./slink_cat readonly
"You can't read!!!"
root@kali:/tmp/test# su - test 사용자 ID변경, 일반 사용자 test 계정으로 변경
$ cd /tmp/test
$ id
uid=1000(test) gid=1001(test) groups=1001(test)
$ ls -alF
합계 12
drwxrwsrwt 2 root root 4096 6월 26 20:08 ./
drwxrwxrwt 9 root root 4096 6월 26 20:13 ../
-r--r----- 1 root root 19 6월 26 20:03 readonly
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
$ ./slink_cat readonly
./slink_cat: readonly: 허가 거부
$ su - root로 사용자 계정 변경
암호 :
root@kali:~# cd /tmp/test
root@kali:/tmp/test# chmod u+s slink_cat setuid 비트 설정
root@kali:/tmp/test# ls -l /bin/cat
-rwsrwxrwx 1 root root 51856 1월 27 2013 /bin/cat
root@kali:/tmp/test# su - test 사용자 ID변경, 일반 사용자 test 계정으로 변경
$ cd /tmp/test
$ ls -alF
합계 12
drwxrwsrwt 2 root root 4096 6월 26 20:08 ./
drwxrwxrwt 9 root root 4096 6월 26 20:14 ../
-r--r----- 1 root root 19 6월 26 20:03 readonly
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
$ ./slink_cat readonly
"You can't read!!!"
$
```

setgid 비트

- setgid 비트: 8진수 2000

- setuid 비트처럼 유효 그룹 ID(EGID)를 사용자의 실제 그룹 ID에서 파일 소유자의 그룹 ID로 변경함

- setgid 비트가 디렉토리에 설정되어 있으면, 이 디렉토리에 새로 설정된 파일들은 디렉토리 그룹 소유권 보다 파일 생성자의 그룹 소유권을 얻게 될 것


```
eunguru — root@kali: /tmp/test — ssh — 80x42
root@kali:/tmp/test# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:/tmp/test# ls -alF
합계 12
drwxrwsrwt 2 root root 4096 6월 26 20:22 ./
drwxrwxrwt 9 root root 4096 6월 26 20:20 ../
-r--r----- 1 root root 19 6월 26 20:03 readonly
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
root@kali:/tmp/test# su - test
$ cd /tmp/test
$ id
uid=1000(test) gid=1001(test) groups=1001(test)
$ umask
0022
$ mkdir sgid_dir
$ touch sgid_file
$ su - root로 사용자 계정 변경
암호:
root@kali:~# cd /tmp/test
root@kali:/tmp/test# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:/tmp/test# ls -alF
합계 16
drwxrwsrwt 3 root root 4096 6월 26 20:23 ./
drwxrwxrwt 9 root root 4096 6월 26 20:20 ../
-r--r----- 1 root root 19 6월 26 20:03 readonly
drwxr-sr-x 2 test root 4096 6월 26 20:23 sgid_dir/
-rw-r--r-- 1 test root 0 6월 26 20:23 sgid_file
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
root@kali:/tmp/test#
```

현재 디렉토리 경로인 /tmp/test 디렉토리에 setgid 비트가 설정되어 있음

사용자 계정 ID 변경, 일반 사용자 test 계정으로 변경

새로운 디렉토리 및 파일 생성

/tmp/test 디렉토리에 setgid 비트가 설정 되기때문에 test 계정에서 생성한 /tmp/test 디렉토리 내의 파일, 디렉토리의 그룹 소유자가 /tmp/test 디렉토리의 그룹 소유자인 root로 되어 있음

- 일반 파일 그룹의 멤버가 파일 소유자의 그룹과 상관없이 디렉토리 내의 모든 파일에 접근이 필요한 공유 디렉토리에 유용

1. setgid 비트 설정방법

- 8진수(2xxx)나 기호(g+s)를 이용하여 setuid 비트를 설정할 수 있음(setgid 비트 설정 제거 g-s)
- 권한 변경을 위해 chmod 명령어를 이용함
- setgid 비트가 설정되어 있으면 그룹 소유자 접근 권한의 실행 권한 자리에 실행 권한이 있으면 소문자 s로 실행권한이 없으면 대문자 S로 표시됨


```
eunguru — root@kali: /home/test2/setgid_dir1 — ssh — 80x42 eunguru
$ id
uid=1001(test2) gid=1002(test2) groups=1002(test2)
$ pwd
/home/test2
$ umask
0022
$ mkdir setgid_dir1 setgid_dir2
$ ls -lF
합계 8
drwxr-xr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir1/
drwxr-xr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir2/
$ chmod 255 setgid_dir1
$ chmod g-x+s setgid_dir2
$ ls -lF
합계 8
drwxr-sr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir1/
drwxr-Sr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir2/
$
$ su - root 사용자 계정으로 변경
암호 :
root@kali:~# cd /home/test2
root@kali:/home/test2# ls -lF
합계 8
drwxr-sr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir1/
drwxr-Sr-x 2 test2 test2 4096 6월 26 22:22 setgid_dir2/
root@kali:/home/test2# cd setgid_dir1
root@kali:/home/test2/setgid_dir1# mkdir dir1
root@kali:/home/test2/setgid_dir1# touch testfile
root@kali:/home/test2/setgid_dir1# ls -alF
합계 12
drwxr-sr-x 3 test2 test2 4096 6월 26 22:26 ./
drwxr-xr-x 4 test2 test2 4096 6월 26 22:22 ../
drwxr-sr-x 2 root test2 4096 6월 26 22:25 dir1/
-rw-r--r-- 1 root test2 0 6월 26 22:26 testfile
root@kali:/home/test2/setgid_dir1#
```

setgid 비트 설정

새로 생성한 디렉토리, 파일의
그룹 소유자가 test2

2. setgid 비트 설정의 활용

1) 사용자 계정 생성시 mail spool 파일 생성

- 사용자 계정 생성시 옵션에 따라 /var/mail 디렉토리 하위에 생성하는 사용자 계정명과 동일명으로 mail spool 파일 생성

- /var/mail 디렉토리에 setgid 비트가 설정되어 있음, 하위에 생성되는 mail spool 파일의 그룹 소유주가 mail이 됨

```
eunguru — root@kali: ~ — ssh — 80x42 eunguru
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~# ls -ld /var/mail
drwxrwsr-x 2 root mail 4096 6월 26 21:39 /var/mail
root@kali:~# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
root@kali:~# useradd -m test2
root@kali:~# ls -ld /var/mail/test2
-rw-rw---- 1 test2 mail 0 6월 26 21:40 /var/mail/test2
root@kali:~#
```

Annotations:

- /var/mail 디렉토리에 setgid 비트 설정 되어 있음, 그룹 소유자는 mail*
- 사용자 계정 생성시 mail spool 파일 생성*
- 사용자 계정 test2 생성시 생성되는 mail spool 파일 그룹 소유자가 test2가 아닌 mail이 됨, /var/mail에 setgid 비트가 설정되어 있기 때문임*

sticky 비트

- sticky 비트: 8진수 1000
- 리눅스는 파일의 sticky bit는 무시
- sticky 비트는 특정 디렉토리를 누구나 자유롭게 사용 할 수 있게 하기 위함 (공용 디렉토리에 사용)
- 단, sticky 비트가 디렉토리에 적용되면 디렉토리 소유자나 파일 소유자 또는 슈퍼유저가 아닌 사용자들은 파일을 삭제하거나 이름을 변경하지 못하도록 막음, 파일 또는 디렉토리 생성은 누구나 할 수 있음
- sticky 비트를 공유모드라고도 함

1. sticky 비트 설정방법 및 활용

- 8진수(1xxx)나 기호(o+t 또는 u+t)를 이용하여 sticky 비트를 설정할 수 있음(sticky 비트 설정 제거 o-t 또는 u-t)
- 리눅스의 경우 o+t, 유닉스(솔라리스)의 경우는 u+t로 설정
- 권한 변경을 위해 chmod 명령어를 이용함
- sticky 비트가 설정되어 있으면 기타 사용자 접근 권한의 실행 권한 자리에 실행 권한이 있으면 소문자 t로 실행권한이 없으면 대문자 T로 표시됨

```
eunguru — root@kali: ~/dir2 — ssh — 80x42 eunguru
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~# umask
0022
root@kali:~# mkdir dir1 dir2 dir3
```

```

root@kali:~# ls -ld dir1 dir2 dir3
drwxr-xr-x 2 root root 4096 6월 26 23:24 dir1
drwxr-xr-x 2 root root 4096 6월 26 23:24 dir2
drwxr-xr-x 2 root root 4096 6월 26 23:24 dir3
root@kali:~# chmod 1754 dir1
root@kali:~# ls -ld dir1
drwxr-xr-t 2 root root 4096 6월 26 23:24 dir1
root@kali:~# chmod 0+t dir2
root@kali:~# ls -ld dir2
drwxr-xr-t 2 root root 4096 6월 26 23:24 dir2
root@kali:~# chmod +t dir3
root@kali:~# ls -ld dir3
drwxr-xr-t 2 root root 4096 6월 26 23:24 dir3

```

sticky 비트 설정

```

root@kali:~#
root@kali:~# chmod 1777 dir2
root@kali:~# cd dir2
root@kali:~/dir2# mkdir dir4
root@kali:~/dir2# touch file1
root@kali:~/dir2# ls -alF

```

sticky비트가 설정되어 있는 dir2
디렉토리 하위에 디렉토리, 파일 생성 가능

```

합계 12
drwxrwxrwt 3 root root 4096 6월 26 23:26 ./
drwxrwxr-x 19 root root 4096 6월 26 23:24 ../
drwxr-xr-x 2 root root 4096 6월 26 23:26 dir4/
-rw-r--r-- 1 root root 0 6월 26 23:26 file1
root@kali:~/dir2# chmod 777 dir4 file1
root@kali:~/dir2# ls -alF
합계 12
drwxrwxrwt 3 root root 4096 6월 26 23:26 ./
drwxrwxr-x 19 root root 4096 6월 26 23:24 ../
drwxrwxrwx 2 root root 4096 6월 26 23:26 dir4/
-rwxrwxrwx 1 root root 0 6월 26 23:26 file1*

```

```

root@kali:~/dir2# su - test
$ cd /root/dir2
$ id
uid=1000(test) gid=1001(test) groups=1001(test)
$

```

test계정으로 사용자 ID 계정 변경

```

$ umask
0022
$ mkdir dir5
$ touch file2

```

sticky 비트가 설정되어 있는
dir2 디렉토리 하위에
디렉토리, 파일 생성 가능

```

$ ls -alF
합계 16
drwxrwxrwt 4 root root 4096 6월 26 23:28 ./
drwxrwxr-x 19 root root 4096 6월 26 23:24 ../
drwxrwxrwx 2 root root 4096 6월 26 23:26 dir4/
drwxr-xr-x 2 test test 4096 6월 26 23:28 dir5/
-rwxrwxrwx 1 root root 0 6월 26 23:26 file1*
-rw-r--r-- 1 test test 0 6월 26 23:28 file2

```

```

$ rmdir dir4
rmdir: failed to remove `dir4': 명령을 허용하지 않음
$ rm file1
rm: cannot remove `file1': 명령을 허용하지 않음
$ rm -rf dir4 file1
rm: cannot remove `dir4': 명령을 허용하지 않음
rm: cannot remove `file1': 명령을 허용하지 않음

```

sticky 비트가 설정되어 있는
dir2 디렉토리 하위의
dir4, file1 삭제 시도
삭제할 수 있는 권한이 있지만 dir2에
sticky비트가 설정되어 있어서 삭제 불가

```

root@kali:~# su - test2
$ cd /root/dir2
$ ls -alF
합계 16
drwxrwxrwt 4 root root 4096 6월 26 23:28 ./
drwxrwxr-x 19 root root 4096 6월 26 23:24 ../
drwxrwxrwx 2 root root 4096 6월 26 23:26 dir4/

```



```
drwxrwxrwt  2 test test 4096  6월 26 23:28 dir5/
-rwxrwxrwx  1 root root   0  6월 26 23:26 file1*
-rw-r--r--  1 test test   0  6월 26 23:28 file2
$ rmdir dir5
rmdir: failed to remove `dir5': 명령을 허용하지 않음
$ su - root 사용자 계정으로 변경
암호 :
root@kali:~# cd /root/dir2
root@kali:~/dir2# ls -alF
합계 16
drwxrwxrwt  4 root root 4096  6월 26 23:28 ./
drwxrwxr-x 19 root root 4096  6월 26 23:24 ../
drwxrwxrwx  2 root root 4096  6월 26 23:26 dir4/
drwxrwxrwt  2 test test 4096  6월 26 23:28 dir5/
-rwxrwxrwx  1 root root   0  6월 26 23:26 file1*
-rw-r--r--  1 test test   0  6월 26 23:28 file2
root@kali:~/dir2# rmdir dir5
root@kali:~/dir2# ls -alF
합계 12
drwxrwxrwt  3 root root 4096  6월 26 23:36 ./
drwxrwxr-x 19 root root 4096  6월 26 23:24 ../
drwxrwxrwx  2 root root 4096  6월 26 23:26 dir4/
-rwxrwxrwx  1 root root   0  6월 26 23:26 file1*
-rw-r--r--  1 test test   0  6월 26 23:28 file2
```

sticky 비트가 설정되어 있는
dir2의 하위 디렉토리나 파일의 삭제는
소유주나 root인 경우에 가능

- /tmp 디렉토리 처럼 공용 디렉토리 접근에 활용

특수 권한 파일 검색

- 특수 권한 비트가 설정되어 있을 때 접근 권한을 이용한 find 명령으로 파일을 검색할 수 있음
- 파일 검색 find 명령에 대한 참조(이전 글 참조)

[2015/05/02 - \[보안/- System\] - \[UNIX / Linux\] 파일 검색\(find\)](#)

1. 특수 권한 비트 설정 파일 검색 시 명령문 형식

- find [파일을 검색할 디렉토리 경로] -perm [접근 권한] [-ls]

1) 파일을 검색할 디렉토리 경로는 생략 가능

- 절대 경로로 지정하면 결과도 절대 경로로 출력, 상대 경로로 지정시 결과도 상대 경로로 출력

- 생략 시 현재 디렉토리가 기준

2) -ls는 생략 가능

-ls 시 검색 결과를 대상으로 ls 명령을 수행한 결과로 보여줌

2. 특수 권한 비트 설정 파일 검색 예제

```
eunguru — root@kali: /tmp/test — ssh — 90x42 eunguru

root@kali:/tmp/test# pwd
/tmp/test
root@kali:/tmp/test# ls -alF
합계 24
drwxrwsrwt 5 root root 4096 6월 27 00:06 ./
drwxrwxrwt 9 root root 4096 6월 27 00:06 ../
drwxr-sr-x 2 root root 4096 6월 27 00:06 normaldir/
-r--r----- 1 root root 19 6월 26 20:03 readonly
drwxr-sr-x 2 test root 4096 6월 26 20:23 sgid_dir/
drwxr-Sr-x 2 test2 root 4096 6월 26 22:14 sgid_dir2/
-rw-r--r-- 1 test root 0 6월 26 20:23 sgid_file
lrwxrwxrwx 1 root root 8 6월 26 20:08 slink_cat -> /bin/cat*
-rwsrwxrwx 1 root root 0 6월 27 00:05 suidfile*
root@kali:/tmp/test#
root@kali:/tmp/test# find /tmp/test -perm -4000 -ls
2111401 4 drwxrwsrwt 5 root root 4096 6월 27 00:06 /tmp/test
2111421 0 -rwsrwxrwx 1 root root 0 6월 27 00:05 /tmp/test/suidfile
root@kali:/tmp/test#
root@kali:/tmp/test# find . -perm -2000 -ls
2111401 4 drwxrwsrwt 5 root root 4096 6월 27 00:06 .
2111422 4 drwxr-sr-x 2 root root 4096 6월 27 00:06 ./normaldir
2111418 4 drwxr-sr-x 2 test root 4096 6월 26 20:23 ./sgid_dir
2111420 4 drwxr-Sr-x 2 test2 root 4096 6월 26 22:14 ./sgid_dir2
root@kali:/tmp/test#
root@kali:/tmp/test# find -perm -1000 -ls
2111401 4 drwxrwsrwt 5 root root 4096 6월 27 00:06 .
root@kali:/tmp/test#
```

setuid 비트 설정 파일 검색

setgid 비트 설정 파일 검색

sticky 비트 설정 파일 검색

신고



'보안 > - System' 카테고리의 다른 글

[UNIX / Linux] 특수 권한(setuid, setgid, sticky bit) (5)	2015.06.27
[UNIX / Linux] 시스템 시작과 종료 (0)	2015.05.10
[UNIX / Linux] 프로세스 응용 (0)	2015.05.10
[UNIX / Linux] 디렉토리 및 파일 관련 명령어 (1)	2015.05.02
[UNIX / Linux] 파일 검색(find) (0)	2015.05.02
[UNIX / Linux] 권한 관리(chmod, chown, chgrp, umask) (10)	2015.04.30
[UNIX / Linux] 디렉토리 관리 (4)	2015.04.29
[UNIX / Linux] 특수 문자(Shell Metacharacter) (0)	2015.04.28
[UNIX / Linux] 파일링크(ln) (0)	2015.04.27
[UNIX / Linux] 입출력 재지정, 파이프 (0)	2015.04.26



Posted by eunguru

TAG [RGID](#), [RUID](#), [setgid](#), [setgid 비트](#), [setgid 비트 설정의 활용](#), [setuid](#), [setuid 비트](#), [setuid 비트 설정의 활용](#), [sticky bit](#), [sticky 비트 설정의 활용](#), [sticky비트](#), [UNIX/Linux 특수권한](#), [\[UNIX / Linux\] 명령어를 찾는 경로와 절차](#), [명령 실행 과정](#), [명령어를 찾는 경로](#), [명령어를 찾는 절차](#), [소유주 권한](#), [실제 사용자 ID](#), [실제 사용자 그룹 ID](#), [유효 사용자 ID](#), [유효 사용자 그룹 ID](#), [특수권한](#), [특수권한 비트 설정 파일 검색](#), [프로세스 번호](#)

* 주의: 전체 웹에서 검색한 결과를 보여줍니다. 블로그 내에서 검색한 결과만 보기를 원한다면 카테고리 위 검색엔진을 이용해주세요.

[트랙백 0개](#), [댓글 5개가 달렸습니다](#)

댓글을 달아 주세요

1. 이민원 2015.08.12 17:37 [신고](#) [댓글주소](#) [수정/삭제](#) [댓글쓰기](#)

안녕하세요. 글 잘 봤습니다.
한 가지 궁금한게 있어서요.

/usr/bin/su 파일에 other 권한을 0(Octal)로 주면 아무리 setuid 설정되어있어도 권한이 없다고 나옵니다. 위에 글을 보면 setuid가 마치 권한이 '없는'사용자가 setuid나 setgid가 설정되어있는경우 그 권한이 effective권한이 되어 실행된단느 내용으로 보여집니다.

하지만 그럴경우 두가지 의문점이 생깁니다.

1. 권한이 없는 사용자가 suid/guid로 실행이 된다고 했는데, owner:group아닌 other 사용자가 suid/guid가 설정되어있는 파일을 실행하고자 할때 실행이 안되는 이유는 먼가요?
2. owner도 group에도 포함되어있지 않은 other 사용자가 suid/guid가 동시에 설정되어있는 경우 suid,guid중 어느 것으로 실행이 되는건가요?

답변 부탁드립니다 ㅌㅌ 감사합니다.

◦ [eunguru](#) 2015.08.13 00:27 [신고](#) [댓글주소](#) [수정/삭제](#)

네. 안녕하세요 ^^
부족한 글에 질문 주신 것을 살펴보았습니다만..

질문에 대한 답변을 드리려면..

질문의 상황에서 사용한 명령, 파일의 접근 권한을 어떻게 설정하셨는지
파일 접근 권한을 기호나 8진수로 기입해서 적어 주셔야
정확한 답변을 드릴 수 있을 것 같습니다.

지금 질문에서 답변을 드리기에는 애매한 부분이 있는 것 같습니다..

추가로 요청 드리는 이유는..

예제를 보통 인위적인 상황을 만들어서 실습해보기 때문에
올바르게 접근 권한을 설정해도 안되는 건지 아니면 실습 상황을 만들면서 잘못 적용
한 부분이 있는지를 확인하기 위함입니다.

물론 저도 혼자 공부하고 이해하고 정리한 내용은 답변을 드릴 실력은 안되지만;;
구체적으로 댓글 남겨주시면 제가 아는한 답변 드리도록 하겠습니다.

2. Q 2015.09.19 14:24 [신고](#) [댓글주소](#) [수정/삭제](#) [댓글쓰기](#)

안녕하세요~ 궁금한 점이 있는데

특정 디렉토리 속성을 selinux에서 웹서버가 접근을 허용하도록 설정하려면 어떻게 해야하나요?

o [eunguru](#) 2015.09.20 01:14 [신고](#) [댓글주소](#) [수정/삭제](#)

이건 제가 정확한 답을 드릴 수 없을 것 같아서 답변 드리지 않겠습니다.

제 답변이 가이드에 맞는 답변인지 증명이 안되서요.

추측성 답변을 드릴수는 없으니까요.

다른 사이트를 검색해보시는게 좋을 것 같네요.

3. [라디아](#) 2016.03.30 21:01 [신고](#) [댓글주소](#) [수정/삭제](#) [댓글쓰기](#)

와 정말 잘 설명해주셨네요. 덕분에 고민해결 ㅋㅋ 감사합니다.

: 이름

: 패스워드

http:// : 홈페이지

☐ 비밀글

댓글 달기



무채색 인간, eunguru

카테고리

- 전체보기 (159)
 - 순간의 기억들. (8)
 - 시간을 보내며 (55)
 - 읽다. (22)
 - 보다. (6)
 - 듣다. (16)
 - 하다. (11)
 - 보안 (21)
 - System (12)
 - Network (3)
 - Web (0)
 - Regulations (2)
 - Etc (3)
 - CISSP (0)
 - CPPG (1)
 - 컴&프로그래밍 (39)
 - Python (17)
 - System Programming (2)
 - C (10)
 - C++ / Win API / MFC (3)
 - Java / JSP (1)

- └─ Node.js (1)
- └─ Database (0)
- └─ Etc, Install (5)
- └─ ● Testing (12)
- └─ ● English (3)
- └─ ● IT 이슈 / 뉴스 (21)

최근에 올라온 글

- [Node.js 설치, hello world 서...](#) (1)
- [CPPG 시험 참고 자료.](#)
- [kali linux에 snort 설치.](#) (3)
- [I \(Feat. 버벌진트\).](#)
- [뷰티 인사이드\(The Beauty Ins...](#)

최근에 달린 댓글

- [좋은 정보 감사합니...](#) Grandpassion 10.31
- [군.](#) 군군 10.24
- [\[승인대기\].](#) 초보자 09.30
- [\[승인대기\].](#) BlackHead 09.28
- [\[승인대기\].](#) fermata39 09.20

태그목록

- [IWELL](#)
- [김성민](#)
- [미 비포 유](#)
- [me before you](#)
- [구글 애드센스 넣기](#)
- [랜섬웨어](#)
- [크립토락커](#)
- [파이썬 연습문제](#)
- [구글 파이썬 튜토리얼](#)
- [튜플](#)
- [장미와 찔레](#)
- [나얼](#)
- [김종완](#)
- [성시경](#)
- [정준일](#)
- [유니코드](#)
- [에버노트 글 불러오기](#)
- [에버노트 티스토리 연동](#)
- [Pocket Evernote 연동](#)
- [Evernote 글 불러오기 플러그인 활용](#)
- [Evernote 글 불러오기 플러그인](#)
- [뉴스기사 Tistory에 스크랩](#)
- [포켓 에버노트 연동](#)
- [뉴스기사 티스토리에 저장](#)

- [Evernote Tistory 연동](#)
- [/etc/passwd](#)
- [≤](#)
- [≥](#)
- [TypeError](#)
- [Else](#)

글 보관함

- [2016/10](#) (1)
- [2016/04](#) (1)
- [2016/03](#) (1)
- [2015/10](#) (1)
- [2015/09](#) (5)

달력

		≪ 2016/11 ≫			
일	월	화	수	목	금 토
		1	2	3	4 5
6	7	8	9	10	11 12
13	14	15	16	17	18 19
20	21	22	23	24	25 26
27	28	29	30		

- Total : 177,283
- Today : 482
- Yesterday : 217



[지역로그](#) : [태그](#) : [미디어로그](#) : [방명록](#) : [관리자](#) : [글쓰기](#)
 eunguru's Blog is powered by [Daum](#) / Designed by [Tistory](#)