Mohammad Arshad

U00304676

NCS 531 11

Linux Capability Exploration Lab

Task 1: Experiencing Capabilities

- Run the command **"ping www.google.com**". From the screenshot we can see that the command is running

```
[02/01/2018 11:02] seed@ubuntu:~$ ping www.google.com
PING www.google.com (173.194.205.103) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping www.google.com
[02/01/2018 11:03] seed@ubuntu:~$ ls -l /bin/ping
-rwsr-xr-x 1 root root 34740 Nov  8  2011 /bin/ping
[02/01/2018 11:03] seed@ubuntu:~$
```

- Changed the **/bin/ping** as a non SETUID program and tried to ping. As seen below the operation was not permitted because ping was not able to open the socket**.**

```
[02/01/2018 11:20] seed@ubuntu:~$
[02/01/2018 11:20] seed@ubuntu:~$ sudo chmod u-s /bin/ping
[sudo] password for seed:
[02/01/2018 11:20] seed@ubuntu:~$ ls -l /bin/ping
-rwxr-xr-x 1 root root 34740 Nov  8  2011 /bin/ping
[02/01/2018 11:21] seed@ubuntu:~$ ping www.google.com
ping: icmp open socket: Operation not permitted
[02/01/2018 11:23] seed@ubuntu:~$
```

- Provided **cap_net_raw** capability to **/bin/ping** and used the ping command. As seen below the command got executed.

```
There are stopped jobs.
[02/01/2018 11:26] seed@ubuntu:~$ su
Password:
[02/01/2018 11:27] root@ubuntu:/home/seed# setcap cap_net_raw=ep /bin/ping
[02/01/2018 11:30] root@ubuntu:/home/seed# exit
exit
[02/01/2018 11:30] seed@ubuntu:~$ ping www.google.com
PING www.google.com (173.194.205.106) 56(84) bytes of data.
^Z
[2]+  Stopped                 ping www.google.com
[02/01/2018 11:30] seed@ubuntu:~$
```

Question 1:

- Changed the **/usr/bin/passwd** to a non SETUID program and executed it. We got the **"Authentication token manipulation error**", the reason is the program is a read only for normal users.

```
[02/12/2018 09:07] seed@ubuntu:~$ /usr/bin/passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
passwd: Authentication token manipulation error
passwd: password unchanged
[02/12/2018 09:07] seed@ubuntu:~$
```

- Provided **cap_dac_override, cap_chown, cap_fowner** capabilities to the **/usr/bin/passwd.** This allowed the successfully updating of password without having root access**.**

```
[02/12/2018 09:07] seed@ubuntu:~$ sudo setcap cap_dac_override,cap_chown,cap_fowner=eip /usr/bin/passwd
[02/12/2018 09:09] seed@ubuntu:~$ /usr/bin/passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
[02/12/2018 09:10] seed@ubuntu:~$
```

Question 2:

a. **CAP_DAC_READ_SEARCH:** Bypass file read permission checks and directory read and execute permission checks

- Created a test file named test.txt and provided no access.

```
[02/11/2018 11:19] seed@ubuntu:~/capab$ ls -l
total 4
---------- 1 root seed 19 Feb 11 11:18 test.txt
```

- Tried to run it using **cat** command but the permission was denied.

```
[02/11/2018 11:20] seed@ubuntu:~/capab$ cat test.txt
cat: test.txt: Permission denied
```

- Provided **CAP_DAC_READ_SEARCH** capability to the /bin/cat program and tried to open the test file using **cat** command and was able to open it.

```
[02/11/2018 11:22] seed@ubuntu:~/capab$ su root
Password:
[02/11/2018 11:23] root@ubuntu:/home/seed/capab# setcap cap_dac_read_search=ep /
bin/cat
[02/11/2018 11:23] root@ubuntu:/home/seed/capab# su seed
[02/11/2018 11:24] seed@ubuntu:~/capab$ ls -l
total 4
---------- 1 root seed 19 Feb 11 11:18 test.txt
-rw-rw-r-- 1 seed seed  0 Feb 11 11:18 Untitled Document
[02/11/2018 11:24] seed@ubuntu:~/capab$ cat test.txt
This is a tes file
[02/11/2018 11:24] seed@ubuntu:~/capab$
```

b. **CAP_DAC_OVERRIDE:** Bypass file read, write, and execute permission checks

- When we type **"vi shadow",** we cannot even open the **/etc/shadow** file.

```
~
~
~
~
~
~
~
~
"/etc/shadow" [Permission Denied]                    0,0-1         All
```

- Provided **cap_dac_override** capability to **/usr/bin/vim.basic** and I was able to open and edit the **etc/shadow** file.

```
[02/11/2018 14:05] seed@ubuntu:/etc$ sudo setcap cap_dac_override=eip /usr/bin/vi
m.basic
[02/11/2018 14:05] seed@ubuntu:/etc$ getcap /usr/bin/vim.basic
/usr/bin/vim.basic = cap_dac_override+eip
[02/11/2018 14:06] seed@ubuntu:/etc$ cd
[02/11/2018 14:06] seed@ubuntu:~$ cd capab
[02/11/2018 14:06] seed@ubuntu:~/capab$ vi /etc/shadow

[2]+  Stopped                 vi /etc/shadow
```

```
bin:*:15749:0:99999:7:::
sys:*:15749:0:99999:7:::
sync:*:15749:0:99999:7:::
games:*:15749:0:99999:7:::
man:*:15749:0:99999:7:::
lp:*:15749:0:99999:7:::
mail:*:15749:0:99999:7:::
news:*:15749:0:99999:7:::
uucp:*:15749:0:99999:7:::
proxy:*:15749:0:99999:7:::
www-data:*:15749:0:99999:7:::
backup:*:15749:0:99999:7:::
list:*:15749:0:99999:7:::
irc:*:15749:0:99999:7:::
gnats:*:15749:0:99999:7:::
nobody:*:15749:0:99999:7:::
libuuid:!:15749:0:99999:7:::
syslog:*:15749:0:99999:7:::
messagebus:*:15749:0:99999:7:::
colord:*:15749:0:99999:7:::
lightdm:*:15749:0:99999:7:::
whoopsie:*:15749:0:99999:7:::
avahi-autoipd:*:15749:0:99999:7:::
avahi:*:15749:0:99999:7:::
usbmux:*:15749:0:99999:7:::
kernoops:*:15749:0:99999:7:::
pulse:*:15749:0:99999:7:::
rtkit:*:15749:0:99999:7:::
speech-dispatcher:!:15749:0:99999:7:::
hplip:*:15749:0:99999:7:::
saned:*:15749:0:99999:7:::
seed:$6$QqXAtWQA$AIjctTUkHMECipE8ElAAJh76YZgrvadHKmWs3hQ3BU8vCC1b5Vv4NhGWw2FsZ01LlZw05L0Gc/p8Plw75hkZR0:15933:0:99999:7:::
mysql:!:15931:0:99999:7:::
bind:*:15931:0:99999:7:::
snort:*:15931:0:99999:7:::
ftp:*:15931:0:99999:7:::
telnetd:*:15931:0:99999:7:::
vboxadd:!:15937::::::
```

c. **CAP_CHOWN:** Make arbitrary changes to file UIDs and GIDs

- Created a test file with owner as root. Tried to change the ownership to user seed being a normal user, as seen below the operation was not permitted. Provided **cap_chown** capability to **/bin/chown** and tried to change the ownership and was successfully changed without root access.

```
[02/11/2018 14:10] seed@ubuntu:~/capab$ chown seed test.txt
chown: changing ownership of `test.txt': Operation not permitted
[02/11/2018 14:11] seed@ubuntu:~/capab$ getcap /bin/chown
[02/11/2018 14:11] seed@ubuntu:~/capab$ sudo setcap cap_chown=eip /bin/chown
[02/11/2018 14:12] seed@ubuntu:~/capab$ chown seed test.txt
[02/11/2018 14:12] seed@ubuntu:~/capab$ ls -l test.txt
---------- 1 seed seed 19 Feb 11 11:18 test.txt
[02/11/2018 14:13] seed@ubuntu:~/capab$
```

d. **CAP_SETUID :** Make arbitrary manipulations of process UIDs

- Created a test file named **test**

```
test
This is a test file for cap_setuid
```

- Created a program that will forge a UID when passing socket.

```
test    testpr.c
#include <unistd.h>
int main()
{
        uid_t uid=0;
        setuid(uid);
        system("/bin/cat /home/seed/test")
        retrun 0;
}
```

- Tried to open the test file but the permission was denied.

```
[02/12/2018 13:38] seed@ubuntu:~$ ls -l test
---------- 1 seed seed 35 Feb 12 13:33 test
[02/12/2018 13:38] seed@ubuntu:~$ cat test
cat: test: Permission denied
[02/12/2018 13:39] seed@ubuntu:~$
```

- Compiled the program and tried to run the program, which also provided the permission denied output.

```
[02/12/2018 13:40] seed@ubuntu:~$ gcc -o testpr testpr.c
[02/12/2018 13:40] seed@ubuntu:~$ ./testpr
/bin/cat: /home/seed/test: Permission denied
[02/12/2018 13:40] seed@ubuntu:~$
```

- Provided **cap_setuid** capability to the executable file testpr and tried running the program. This time the test file was open. The **cap_setuid** capability allowed manipulations of process UIDs.

```
[02/12/2018 13:40] seed@ubuntu:~$ sudo setcap cap_setuid=eip ./testpr
[02/12/2018 13:42] seed@ubuntu:~$ ./testpr
This is a test file for cap_setuid
[02/12/2018 13:42] seed@ubuntu:~$
```

e. CAP_KILL : Bypass permission checks for sending signals

- Executed the **top** command in another terminal as root user.

```
Tasks: 161 total,   1 running, 160 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.7%us,  1.0%sy,  0.0%ni, 97.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   2064788k total,  1203016k used,   861772k free,   138572k buffers
Swap:  2094076k total,        0k used,  2094076k free,   622548k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
12293 root      20   0 74760  41m  10m S  1.0  2.1   0:06.64 Xorg
12845 seed      20   0 90428  15m  10m S  0.7  0.7   0:01.50 gnome-terminal
13092 root      20   0  2852 1160  876 R  0.3  0.1   0:00.15 top
    1 root      20   0  3672 2072 1288 S  0.0  0.1   0:01.60 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:01.13 ksoftirqd/0
    5 root      20   0     0    0    0 S  0.0  0.0   0:02.00 kworker/u:0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    7 root      RT   0     0    0    0 S  0.0  0.0   0:00.67 watchdog/0
    8 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 cpuset
    9 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kdevtmpfs
   11 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.10 sync_supers
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.00 bdi-default
   14 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kintegrityd
   15 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kblockd
```

- Tried to Kill the process using **/bin/kill**, but was not able to do. Provided **cap_kill** capability to **/bin/kill** and tried to kill the process. As shown below I was able to kill the top process of root user without accessing the root.

```
[02/11/2018 14:45] seed@ubuntu:~$ pgrep top
13092
[02/11/2018 14:45] seed@ubuntu:~$ /bin/kill 13092
kill: Operation not permitted
[02/11/2018 14:45] seed@ubuntu:~$ sudo setcap cap_kill=eip /bin/kill
[02/11/2018 14:46] seed@ubuntu:~$ /bin/kill 13092
[02/11/2018 14:46] seed@ubuntu:~$
```

f. CAP_NET_RAW: use RAW and PACKET sockets

- Run the command **"ping www.google.com**". From the screenshot we can see that the command is running

```
[02/01/2018 11:02] seed@ubuntu:~$ ping www.google.com
PING www.google.com (173.194.205.103) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping www.google.com
[02/01/2018 11:03] seed@ubuntu:~$ ls -l /bin/ping
-rwsr-xr-x 1 root root 34740 Nov  8  2011 /bin/ping
[02/01/2018 11:03] seed@ubuntu:~$
```

- Changed the **/bin/ping** as a non SETUID program and tried to ping. As seen below the operation was not permitted because ping was not able to open the socket**.**

```
[02/01/2018 11:20] seed@ubuntu:~$
[02/01/2018 11:20] seed@ubuntu:~$ sudo chmod u-s /bin/ping
[sudo] password for seed:
[02/01/2018 11:20] seed@ubuntu:~$ ls -l /bin/ping
-rwxr-xr-x 1 root root 34740 Nov  8  2011 /bin/ping
[02/01/2018 11:21] seed@ubuntu:~$ ping www.google.com
ping: icmp open socket: Operation not permitted
[02/01/2018 11:23] seed@ubuntu:~$
```

- Provided **cap_net_raw** capability to **/bin/ping** and used the ping command. As seen below the command got executed.

```
[02/01/2018 11:26] seed@ubuntu:~$ su
Password:
[02/01/2018 11:27] root@ubuntu:/home/seed# setcap cap_net_raw=ep /bin/ping
[02/01/2018 11:30] root@ubuntu:/home/seed# exit
exit
[02/01/2018 11:30] seed@ubuntu:~$ ping www.google.com
PING www.google.com (173.194.205.106) 56(84) bytes of data.
^Z
[2]+  Stopped                 ping www.google.com
[02/01/2018 11:30] seed@ubuntu:~$
```

Task 2: Adjusting Privileges

Added the given functions to **cap_prog.c** and compiled and installed the updated **libcap.**

- Compiled the **use_cap.c** program

```
[02/12/2018 12:27] root@ubuntu:/home/seed# gcc -c use_cap.c
[02/12/2018 12:27] root@ubuntu:/home/seed# gcc -o use_cap use_cap.o -lcap
```

- Provided **cap_dac_read_search** capability to the executable file of **use_cap.c**.
  Run the program in root user to get the below output.

```
[02/12/2018 12:29] root@ubuntu:/home/seed# setcap cap_dac_read_search=eip use_ca
p
[02/12/2018 12:29] root@ubuntu:/home/seed# su seed
[02/12/2018 12:30] seed@ubuntu:~$ ./use_cap
(b) Open failed
(d) Open failed
(e) Open failed
[02/12/2018 12:30] seed@ubuntu:~$
```

Compare with the code and the result,

- First the program cannot open shadow file, it comes **(b) open failed,**
- After that we enable the capability , it can get the capability and bypassed the **(c) open failed**,

- Next we drop the capability, and printed **(d)open failed**,
- We cannot enable it again after dropping the capability, though, in the code, even we tried to enable its capability again, so printed **(e) open failed**.

If we change the code a little bit

if (cap_enable(CAP_DAC_READ_SEARCH) < 0) return -1;

if (open ("/etc/shadow", O_RDONLY) > 0)

printf("(c) Open Ssccesful\n");

We will get the below output adding "**(c) Open successful**"

```
[02/12/2018 12:42] seed@ubuntu:~$ ./use_cap
(b) Open failed
(c) Open successful
(d) Open failed
(e) Open failed
[02/12/2018 12:42] seed@ubuntu:~$
```

Question 4:

ACL is a list of access control entry, which give access permission to a user or group on a given file or folder. In ACL, if we want to grant permission to other user/group, we always need to login as root or superuser, and use "chmod" command to grand permission on file to the aimed user. While by using capabilities, we can bypass some permission check, even if we were not supposed to have permission on accessing this file. It is convenient for normal user since you do not need to ask access permission from root, but it is more problematic considering the security side.

Question 5:

Yes. After normal user disables a capability A, the attacker can still use the capability A by enabling it in his malicious code, but if the process deleted the capability, the attacker cannot use the capability.

Question 6:

If the attacker exploits the race condition in this program, he can still use the capability A no matter the capability is disabled or deleted. That is because, in the race condition attack the malicious code will always run before the capability statement.

Reference:

http://www.cis.syr.edu/~wedu/Teaching/IntrCompSec/LectureNotes_New/Race_Condition.pdf