Set-UID与Capability

1.解释“passwd”，“sudo”，“ping”等命令为什么需要setuid位，去掉s位试运行，添加权能试运行。

答：“passwd”，”sudo”，”ping”都需要root权限来运行命令，普通用户没有权限运行，设置setuid位，任何用户执行时，都以setuid程序文件所属的用户的身份运行。

实验过程如下：

(1) passwd

去掉s位后无法执行：

```
zmb@ubuntu:~$ ll $(which passwd)
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd*
zmb@ubuntu:~$ sudo chmod u-s $(which passwd)
zmb@ubuntu:~$ ll $(which passwd)
-rwxr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd*
zmb@ubuntu:~$ passwd
Changing password for zmb.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: Authentication token manipulation error
passwd: password unchanged
```

添加相应权能后恢复执行：

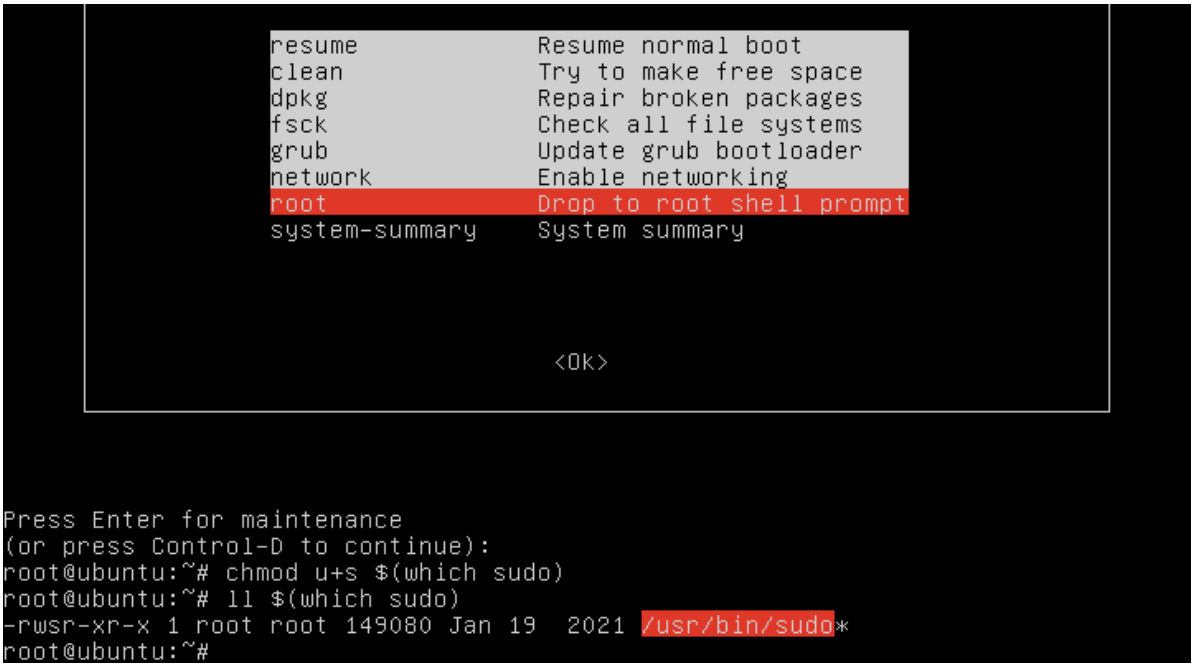
```
zmb@ubuntu:~$ ll $(which passwd)
-rwxr-xr-x 1 root root 59640 Nov 29 04:25 /usr/bin/passwd*
zmb@ubuntu:~$ getcap $(which passwd)
zmb@ubuntu:~$ sudo setcap 'cap_fowner+ep cap_chown+ep cap_dac_override+ep' $(wh
ich passwd)
zmb@ubuntu:~$ getcap $(which passwd)
/usr/bin/passwd = cap_chown,cap_dac_override,cap_fowner+ep
zmb@ubuntu:~$ passwd
Changing password for zmb.
(current) UNIX password:
Enter new UNIX password:
```

(2) sudo

去掉s位后无法执行：

```
zmb@ubuntu:~$ ll $(which sudo)
-rwsr-xr-x 1 root root 149080 Jan 19  2021 /usr/bin/sudo*
zmb@ubuntu:~$ sudo chmod u-s $(which sudo)
[sudo] password for zmb:
zmb@ubuntu:~$ ll $(which sudo)
-rwxr-xr-x 1 root root 149080 Jan 19  2021 /usr/bin/sudo*
zmb@ubuntu:~$ sudo apt-get install gcc
sudo: /usr/bin/sudo must be owned by uid 0 and have the setuid bit set
```

恢复s位：



(3) ping

去掉s位后无法执行：

```
zmb@ubuntu:~$ ll $(which ping)
-rwsr-xr-x 1 root root 64424 Jun 28  2019 /bin/ping*
zmb@ubuntu:~$ ping baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=128 time=8.62 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=128 time=8.18 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=128 time=9.45 ms
^C
--- baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 8.186/8.753/9.451/0.524 ms
zmb@ubuntu:~$ sudo chmod u-s $(which ping)
[sudo] password for zmb:
zmb@ubuntu:~$ ll $(which ping)
-rwxr-xr-x 1 root root 64424 Jun 28  2019 /bin/ping*
zmb@ubuntu:~$ ping baidu.com
ping: socket: Operation not permitted
```

添加相应权能后恢复执行：

```
zmb@ubuntu:~$ ll $(which ping)
-rwxr-xr-x 1 root root 64424 Jun 28  2019 /bin/ping*
zmb@ubuntu:~$ getcap $(which ping)
zmb@ubuntu:~$ sudo setcap cap_net_raw+ep $(which ping)
zmb@ubuntu:~$ getcap $(which ping)
/bin/ping = cap_net_raw+ep
zmb@ubuntu:~$ ping baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=128 time=9.35 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=128 time=9.50 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=128 time=9.50 ms
^C
--- baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 9.357/9.455/9.508/0.069 ms
```

2.指出每个权能对应的系统调用，简要解释功能

通过find命令查找capability.h所在的位置，查看权能详情：

```
zmb@ubuntu:~$ sudo find / -name capability.h
[sudo] password for zmb:
find: ‘/run/user/1000/gvfs’: Permission denied
find: ‘/run/user/121/gvfs’: Permission denied
/usr/include/linux/capability.h
/usr/src/linux-hwe-5.4-headers-5.4.0-84/include/uapi/linux/capability.h
/usr/src/linux-hwe-5.4-headers-5.4.0-84/include/linux/capability.h
/usr/src/linux-hwe-5.4-headers-5.4.0-144/include/uapi/linux/capability.h
/usr/src/linux-hwe-5.4-headers-5.4.0-144/include/linux/capability.h
zmb@ubuntu:~$ vi /usr/src/linux-hwe-5.4-headers-5.4.0-84/include/uapi/linux/cap
ability.h
```

总结如下：

权能	编号	描述
CAP_CHOWN	0(chown)	允许改变文件的所有权
CAP_DAC_OVERRIDE	1	忽略对文件的所有DAC访问限制
CAP_DAC_READ_SEARCH	2	忽略所有对读、搜索操作的限制
CAP_FOWNER	3	如果文件属于进程的UID，就取消对文件的限制
CAP_FSETID	4	允许设置setuid位
CAP_KILL	5(kill)	允许对不属于自己的进程发送信号

权能	编号	描述
CAP_SETGID	6(setgid)	允许改变组ID
CAP_SETUID	7(setuid)	允许改变用户ID
CAP_SETPCAP	8(capset)	允许向其它进程转移能力以及删除其它进程的任意能力
CAP_LINUX_IMMUTABLE	9(chattr)	允许修改文件的不可修改(IMMUTABLE)和只添加(APPEND-ONLY)属性
CAP_NET_BIND_SERVICE	10	允许绑定到小于1024的端口
CAP_NET_BROADCAST	11	允许网络广播和多播访问
CAP_NET_ADMIN	12	允许执行网络管理任务：接口、防火墙和路由等
CAP_NET_RAW	13(socket)	允许使用原始(raw)套接字
CAP_IPC_LOCK	14(mlock)	允许锁定共享内存片段
CAP_IPC_OWNER	15	忽略IPC所有权检查
CAP_SYS_MODULE	16(init_module)	插入和删除内核模块
CAP_SYS_RAWIO	17	允许对ioperm/iopl的访问
CAP_SYS_CHROOT	18(chroot)	允许使用chroot()系统调用
CAP_SYS_PTRACE	19(ptrace)	允许跟踪任何进程
CAP_SYS_PACCT	20(acct)	允许配置进程记帐(process accounting)
CAP_SYS_ADMIN	21	允许执行系统管理任务：加载/卸载文件系统、设置磁盘配额、开/关交换设备和文件等
CAP_SYS_BOOT	22(reboot)	允许重新启动系统
CAP_SYS_NICE	23(nice)	允许提升优先级，设置其它进程的优先级
CAP_SYS_RESOURCE	24(setrlimit)	忽略资源限制
CAP_SYS_TIME	25(stime)	允许改变系统时钟
CAP_SYS_TTY_CONFIG	26(vhangup)	允许配置TTY设备
CAP_MKNOD	27(mknod)	允许使用mknod()系统调用
CAP_LEASE	28(fcntl)	为任意文件建立租约
CAP_AUDIT_WRITE	29	允许向内核审计日志写记录
CAP_AUDIT_CONTROL	30	启用或禁用内核审计，修改审计过滤器规则
CAP_SETFCAP	31	设置文件权能
CAP_MAC_OVERRIDE	32	允许MAC配置或状态改变，为smack LSM实现

权能	编号	描述
CAP_MAC_ADMIN	33	覆盖强制访问控制
CAP_SYSLOG	34(syslog)	执行特权syslog(2)操作
CAP_WAKE_ALARM	35	触发将唤醒系统的东西
CAP_BLOCK_SUSPEND	36(epoll)	可以阻塞系统挂起的特性
CAP_AUDIT_READ	37	允许通过一个多播socket读取审计日志

3.查找你Linux发行版系统(Ubuntu/centos等)中所有设置了setuid位的程序，指出其应该有的权能

使用`sudo find / -perm /u=s`命令查找所有设置了setuid位的程序，结果如下：

```
/usr/lib/openssh/ssh-keysign
/usr/lib/snapd/snap-confine
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/sbin/pppd
/usr/bin/gpasswd
/usr/bin/arping
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/vmware-user-suid-wrapper
/usr/bin/sudo
/usr/bin/traceroute6.iputils
/snap/core20/1081/usr/bin/chfn
/snap/core20/1081/usr/bin/chsh
/snap/core20/1081/usr/bin/gpasswd
/snap/core20/1081/usr/bin/mount
/snap/core20/1081/usr/bin/newgrp
/snap/core20/1081/usr/bin/passwd
/snap/core20/1081/usr/bin/su
/snap/core20/1081/usr/bin/sudo
/snap/core20/1081/usr/bin/umount
/snap/core20/1081/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core20/1081/usr/lib/openssh/ssh-keysign
/snap/core20/1828/usr/bin/chfn
/snap/core20/1828/usr/bin/chsh
/snap/core20/1828/usr/bin/gpasswd
/snap/core20/1828/usr/bin/mount
/snap/core20/1828/usr/bin/newgrp
/snap/core20/1828/usr/bin/passwd
/snap/core20/1828/usr/bin/su
/snap/core20/1828/usr/bin/sudo
/snap/core20/1828/usr/bin/umount
/snap/core20/1828/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

```
/snap/core20/1828/usr/lib/openssh/ssh-keysign
/snap/snapd/18357/usr/lib/snapd/snap-confine
/snap/core18/2708/bin/mount
/snap/core18/2708/bin/ping
/snap/core18/2708/bin/su
/snap/core18/2708/bin/umount
/snap/core18/2708/usr/bin/chfn
/snap/core18/2708/usr/bin/chsh
/snap/core18/2708/usr/bin/gpasswd
/snap/core18/2708/usr/bin/newgrp
/snap/core18/2708/usr/bin/passwd
/snap/core18/2708/usr/bin/sudo
/snap/core18/2708/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/2708/usr/lib/openssh/ssh-keysign
/snap/core18/2128/bin/mount
/snap/core18/2128/bin/ping
/snap/core18/2128/bin/su
/snap/core18/2128/bin/umount
/snap/core18/2128/usr/bin/chfn
/snap/core18/2128/usr/bin/chsh
/snap/core18/2128/usr/bin/gpasswd
/snap/core18/2128/usr/bin/newgrp
/snap/core18/2128/usr/bin/passwd
/snap/core18/2128/usr/bin/sudo
/snap/core18/2128/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/2128/usr/lib/openssh/ssh-keysign
/snap/core22/522/usr/bin/chfn
/snap/core22/522/usr/bin/chsh
/snap/core22/522/usr/bin/gpasswd
/snap/core22/522/usr/bin/mount
/snap/core22/522/usr/bin/newgrp
/snap/core22/522/usr/bin/passwd
/snap/core22/522/usr/bin/su
/snap/core22/522/usr/bin/sudo
/snap/core22/522/usr/bin/umount
/snap/core22/522/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core22/522/usr/lib/openssh/ssh-keysign
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
```

部分程序及其对应的权能如下：

程序	权能	描述
/bin/su	CAP_DAC_OVERRIDE	忽略对文件的所有DAC访问限制
	CAP_SETGID	允许改变组ID
	CAP_SETUID	允许改变用户ID
/bin/ping	CAP_NET_RAW	允许使用原始(raw)套接字
/usr/bin/passwd	CAP_CHOWN	允许改变文件的所有权
	CAP_DAC_OVERRIDE	忽略对文件的所有DAC访问限制
	CAP_FOWNER	如果文件属于进程的UID，就取消对文件的限制
/bin/fusermount	CAP_SYS_ADMIN	允许执行系统管理任务：加载/卸载文件系统、设置磁盘配额、开/关交换设备和文件等
/bin/umount	CAP_SYS_ADMIN	允许执行系统管理任务：加载/卸载文件系统、设置磁盘配额、开/关交换设备和文件等
/bin/mount	CAP_SYS_ADMIN	允许执行系统管理任务：加载/卸载文件系统、设置磁盘配额、开/关交换设备和文件等

4.实现一个程序其满足以下的功能：

(1)能够永久的删除其子进程的某个权能。

(2)能暂时性的删除其子进程的某个权能。

(3)能让上面被暂时性删除的权能重新获得。

常规功能测试

使用`sudo`执行程序后设置权能，并在成功切换成普通用户与普通组时传递权能，进行ping测试：

```
zmb@ubuntu:~/Documents$ sudo ./capability_test3
[sudo] password for zmb:
root用户下的进程和特权信息：
当前进程 (uid=0, euid=0, gid=0)
当前进程pid=110897, 权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+ep
普通用户下的进程特权信息：
当前进程 (uid=1000, euid=1000, gid=1000)
当前进程pid=110897, 权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+eip
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
5
执行shell命令的子进程 (uid=1000, euid=1000, gid=1000)
执行shell命令的子进程pid=110910, 权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+eip
进行ping baidu.com测试：
PING baidu.com (110.242.68.66) 56(84) bytes of data.
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=128 time=17.4 ms

--- baidu.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 17.443/17.443/17.443/0.000 ms
```

暂时删除cap_net_raw权能后进行ping操作失败：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
2
请输入想要操作的权能名称，如cap_net_raw（不区分大小写）：
cap_net_raw
【info】：成功暂时删除进程的cap_net_raw权能
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
4
当前进程（uid=1000，euid=1000，gid=1000）
当前进程pid=110897，权能= cap_setgid,cap_setuid,cap_setpcap+eip cap_net_raw+p
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
5
执行shell命令的子进程（uid=1000，euid=1000，gid=1000）
执行shell命令的子进程pid=110925，权能= cap_setgid,cap_setuid,cap_setpcap+eip cap_net_raw+p
进行ping baidu.com测试：
ping: socket: Operation not permitted
```

恢复cap_net_raw权能后ping操作成功：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
3
【info】：成功恢复进程被暂时删除的权能
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
4
当前进程（uid=1000，euid=1000，gid=1000）
当前进程pid=110897，权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+eip
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
5
执行shell命令的子进程（uid=1000，euid=1000，gid=1000）
执行shell命令的子进程pid=110929，权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+eip
进行ping baidu.com测试：
PING baidu.com (110.242.68.66) 56(84) bytes of data.
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=128 time=18.3 ms

--- baidu.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 18.375/18.375/18.375/0.000 ms
```

永久删除cap_net_raw权能后ping操作失败：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
1
请输入想要操作的权能名称，如cap_net_raw（不区分大小写）：
cap_net_raw
【info】：成功永久删除进程的cap_net_raw权能
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
4
当前进程（uid=1000，euid=1000，gid=1000）
当前进程pid=110897，权能= cap_setgid,cap_setuid,cap_setpcap+eip
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
5
执行shell命令的子进程（uid=1000，euid=1000，gid=1000）
执行shell命令的子进程pid=110945，权能= cap_setgid,cap_setuid,cap_setpcap+eip
进行ping baidu.com测试：
ping: socket: Operation not permitted
```

尝试恢复被永久删除的cap_net_raw权能失败：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
3
【error】：设置当前进程的权能状态失败！
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
4
当前进程（uid=1000，euid=1000，gid=1000）
当前进程pid=110897，权能= cap_setgid,cap_setuid,cap_setpcap+eip
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
5
执行shell命令的子进程（uid=1000，euid=1000，gid=1000）
执行shell命令的子进程pid=110948，权能= cap_setgid,cap_setuid,cap_setpcap+eip
进行ping baidu.com测试：
ping: socket: Operation not permitted
```

异常功能测试

未使用`sudo`执行程序：

```
zmb@ubuntu:~/Documents$ ./capability_test3
请使用sudo执行本程序
```

刚进入程序就尝试恢复：

```
zmb@ubuntu:~/Documents$ sudo ./capability_test3
[sudo] password for zmb:
root用户下的进程和特权信息：
当前进程（uid=0，euid=0，gid=0）
当前进程pid=110954，权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+ep
普通用户下的进程特权信息：
当前进程（uid=1000，euid=1000，gid=1000）
当前进程pid=110954，权能= cap_setgid,cap_setuid,cap_setpcap,cap_net_raw+eip
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
3
【info】：没有可以恢复的权能！
```

输入选项错误：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
a
【warning】：请输入正确的数字或符号！
```

权能名称错误：

```
 1：永久地删除子进程的某个权能。
 2：暂时性地删除子进程的某个权能。
 3：让上面被暂时性删除的权能重新获得。
 4：查看当前进程的权能
 5：启用子进程进行ping操作
 q：退出本程序。
 请输入选项对应的数字并回车：
2
请输入想要操作的权能名称，如cap_net_raw（不区分大小写）：
aaaaaaaaa
【warning】：请输入正确的权能名称！
```