Experiment Report

Name 吳彬睿 Student ID 3180200084

Exp. Title Enviroment variable and set-uid lab Exp. Date 2019/5/7

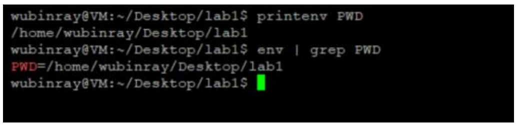
一、Basic Principles (原理简述)

本实验的学习目标是让学生了解环境变量如何影响程序和系统行为。 环境变量是一组可影响方式的动态命名值正在运行的进程将在计算机上运行。 它们被大多数操作系统使用，因为它们是尽管环境变量会影响程序行为，但它们是如何实现的许多程序员都不太了解这一点。 因此，如果程序使用环境变量，但程序员不知道他们被使用，程序可能有漏洞。 在这个实验室里学生将了解环境变量如何工作，如何从父进程传播到孩子，以及它们如何影响系统/程序行为。 我们对环境如何特别感兴趣变量影响 Set-UID 程序的行为，这些程序通常是特权程序。

Report due Mon 11:59pm to MobileSecurity2014@163.com. You can write in either English or Chinese. Document naming convention: ExperimentDate- ChineseName-StudentID, e.g., 2014-11-24-张三-123456789.

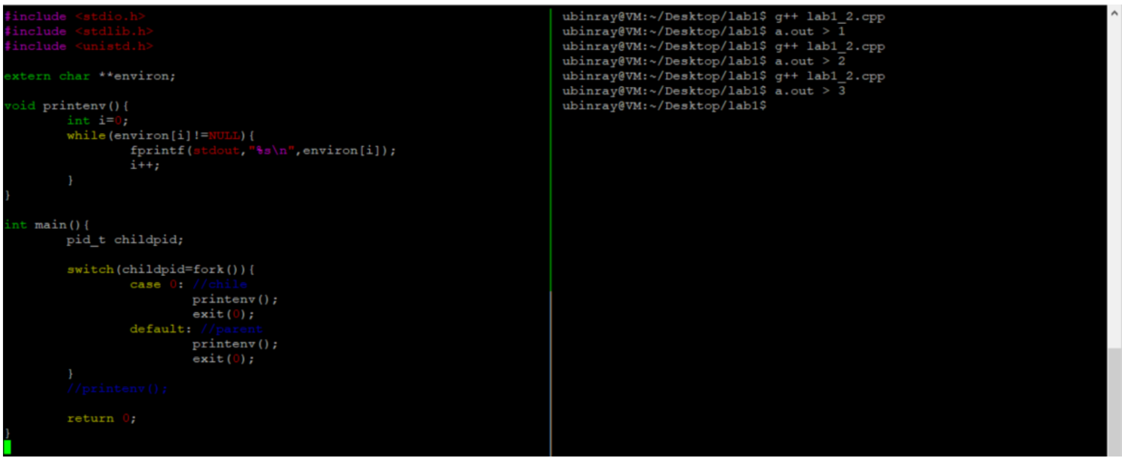
二、Step-by-Step Procedure (实验步骤)

(一) Manipulate Enviroment Variables :



```
wubinray@VM:~/Desktop/lab1$ printenv PWD
/home/wubinray/Desktop/lab1
wubinray@VM:~/Desktop/lab1$ env | grep PWD
PWD=/home/wubinray/Desktop/lab1
wubinray@VM:~/Desktop/lab1$ 
```

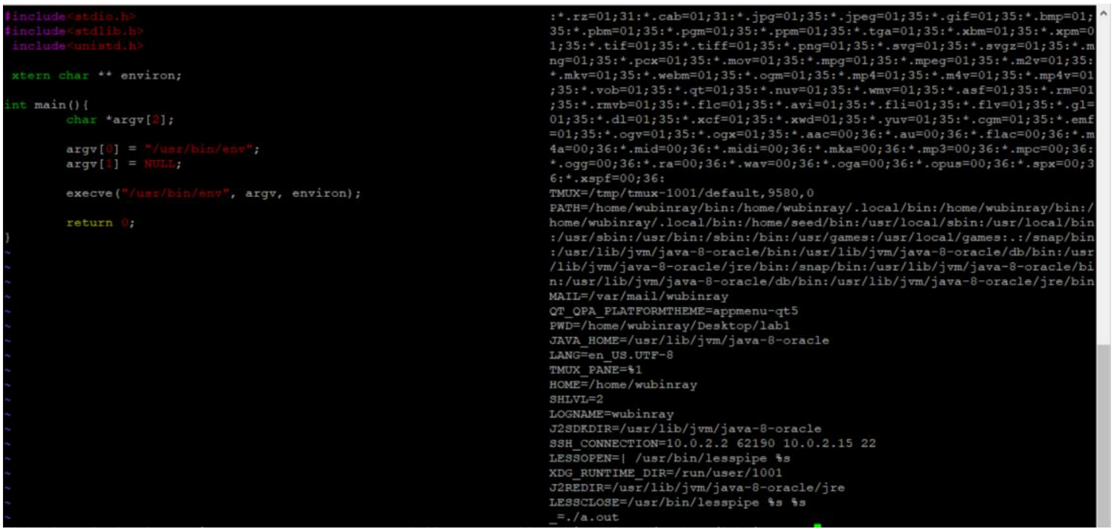
(二) Passing Environment Variables from Parent Process to Child Process:



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

extern char **environ;

void printenv(){
        int i=0;
        while(environ[i]!=NULL){
                fprintf(stdout,"%s\n",environ[i]);
                i++;
        }
}

int main(){
        pid_t childpid;

        switch(childpid=fork()){
                case 0: //chile
                        printenv();
                        exit(0);
                default: //parent
                        printenv();
                        exit(0);
        }
        //printenv();

        return 0;
}
```

```
ubinray@VM:~/Desktop/lab1$ g++ lab1_2.cpp
ubinray@VM:~/Desktop/lab1$ a.out > 1
ubinray@VM:~/Desktop/lab1$ g++ lab1_2.cpp
ubinray@VM:~/Desktop/lab1$ a.out > 2
ubinray@VM:~/Desktop/lab1$ g++ lab1_2.cpp
ubinray@VM:~/Desktop/lab1$ a.out > 3
ubinray@VM:~/Desktop/lab1$
```

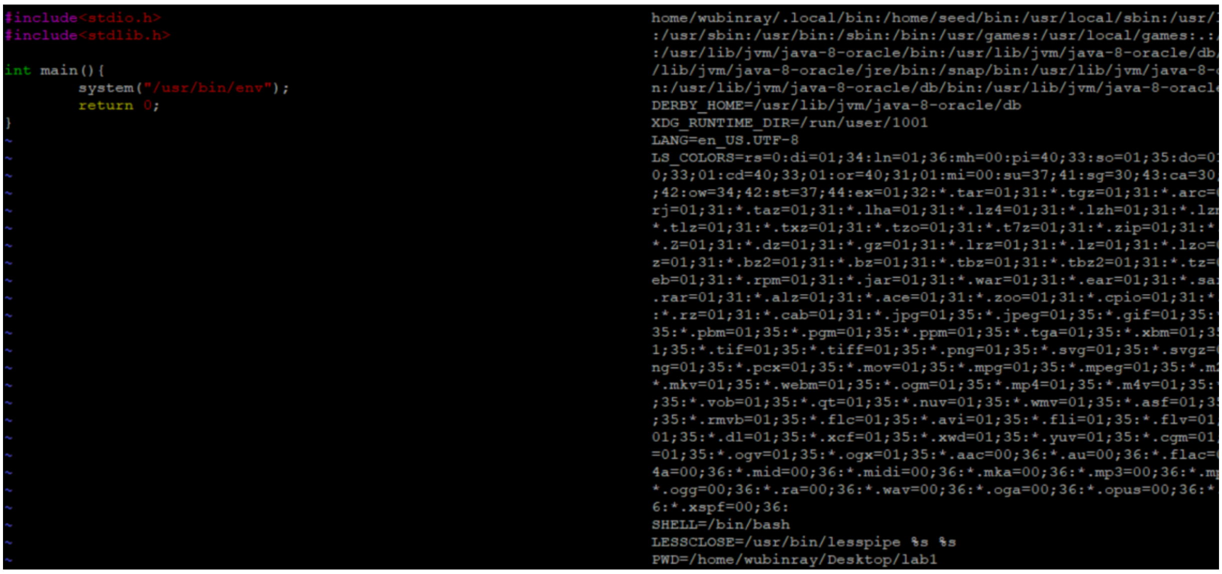
(三) Environment Variables and execve() :



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

 xtern char ** environ;

int main(){
       char *argv[2];

       argv[0] = "/usr/bin/env";
       argv[1] = NULL;

       execve("/usr/bin/env", argv, environ);

       return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.m
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;3
6:*.xspf=00;36:
TMUX=/tmp/tmux-1001/default,9580,0
PATH=/home/wubinray/bin:/home/wubinray/.local/bin:/home/wubinray/bin:/
home/wubinray/.local/bin:/home/seed/bin:/usr/local/sbin:/usr/local/bin
:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin
:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr
/lib/jvm/java-8-oracle/jre/bin:/snap/bin:/usr/lib/jvm/java-8-oracle/bi
n:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin
MAIL=/var/mail/wubinray
QT_QPA_PLATFORMTHEME=appmenu-qt5
PWD=/home/wubinray/Desktop/lab1
JAVA_HOME=/usr/lib/jvm/java-8-oracle
LANG=en_US.UTF-8
TMUX_PANE=%1
HOME=/home/wubinray
SHLVL=2
LOGNAME=wubinray
J2SDKDIR=/usr/lib/jvm/java-8-oracle
SSH_CONNECTION=10.0.2.2 62190 10.0.2.15 22
LESSOPEN=| /usr/bin/lesspipe %s
XDG_RUNTIME_DIR=/run/user/1001
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
_=./a.out
```

(四) Enviroment Variables and system() :

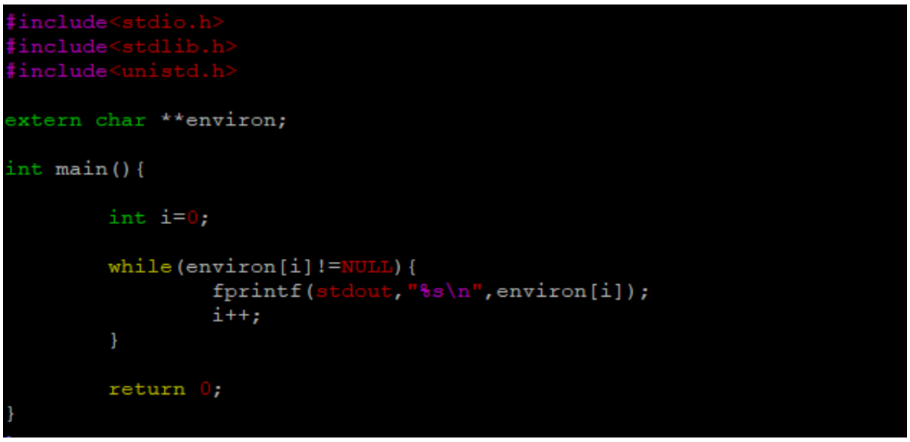


```
#include<stdio.h>
#include<stdlib.h>

int main(){
        system("/usr/bin/env");
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
home/wubinray/.local/bin:/home/seed/bin:/usr/local/sbin:/usr/
:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:
:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db
/lib/jvm/java-8-oracle/jre/bin:/snap/bin:/usr/lib/jvm/java-8-
n:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracl
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
XDG_RUNTIME_DIR=/run/user/1001
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=0
0;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30
;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=
rj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lz
*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*
*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=
z=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sa
.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;3
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;3
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.m
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*
6:*.xspf=00;36:
SHELL=/bin/bash
LESSCLOSE=/usr/bin/lesspipe %s %s
PWD=/home/wubinray/Desktop/lab1
```

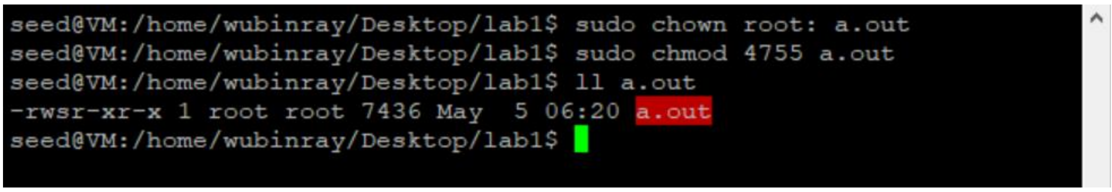
(五) Enviroment Variable and Set-UID Programs :

(1) 写一个会 print 环境变数的程序



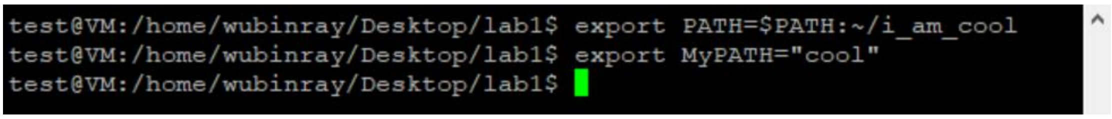
```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

extern char **environ;

int main(){

        int i=0;

        while(environ[i]!=NULL){
                fprintf(stdout,"%s\n",environ[i]);
                i++;
        }

        return 0;
}
```

(2)编译挡案，并更改执行挡案的权限以及持有者



```
seed@VM:/home/wubinray/Desktop/lab1$ sudo chown root: a.out
seed@VM:/home/wubinray/Desktop/lab1$ sudo chmod 4755 a.out
seed@VM:/home/wubinray/Desktop/lab1$ ll a.out
-rwsr-xr-x 1 root root 7436 May  5 06:20 a.out
seed@VM:/home/wubinray/Desktop/lab1$ 
```

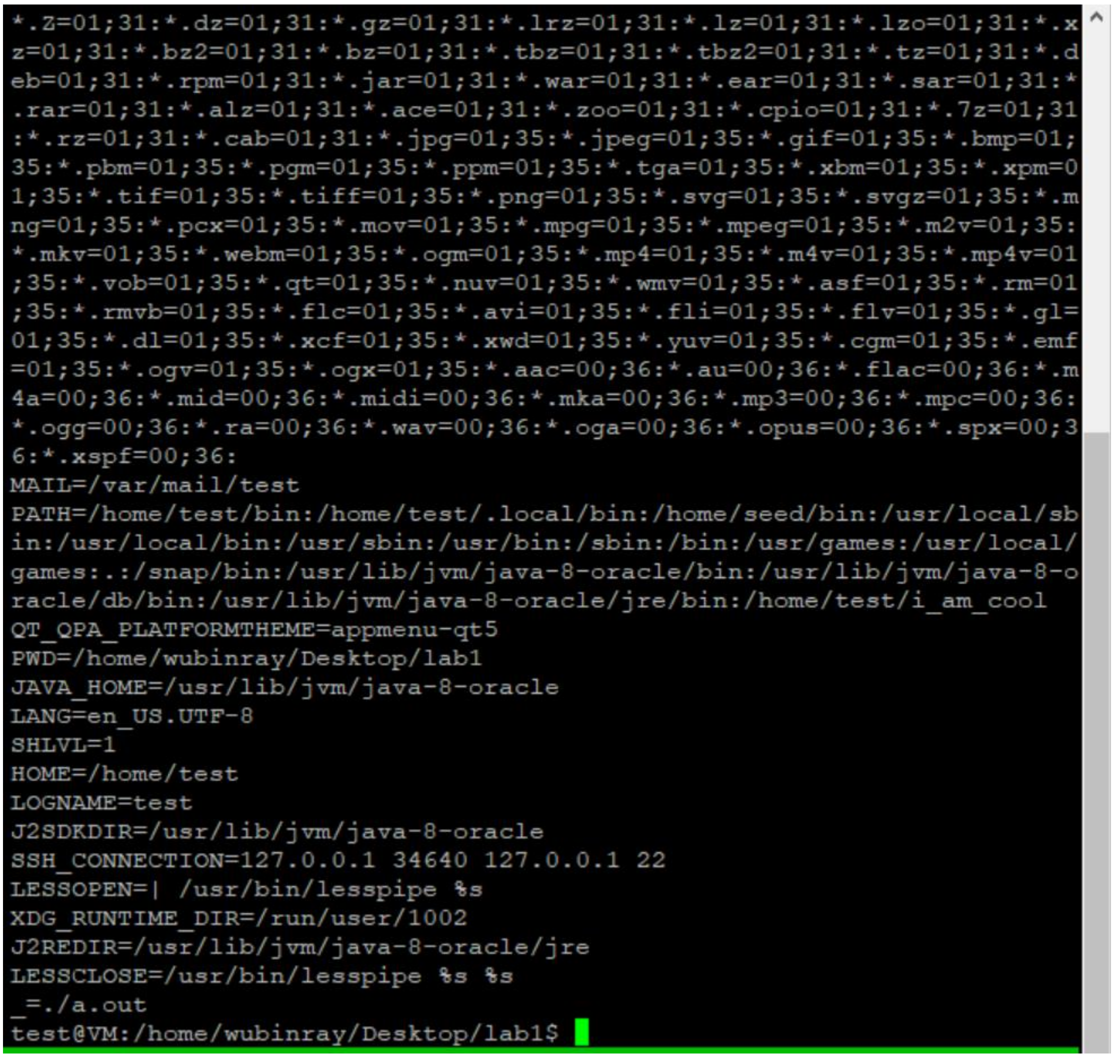
(3)登入一个非 root 的账号，并更改环境变数



```
test@VM:/home/wubinray/Desktop/lab1$ export PATH=$PATH:~/i_am_cool
test@VM:/home/wubinray/Desktop/lab1$ export MyPATH="cool"
test@VM:/home/wubinray/Desktop/lab1$ 
```

(4)执行刚刚写好的程序，看看 print 出来的环境变数是否有变化

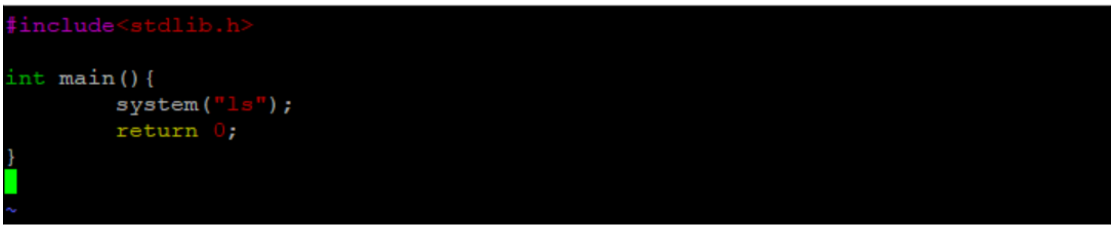
Report due Mon 11:59pm to MobileSecurity2014@163.com. You can write in either English or Chinese. Document naming convention: ExperimentDate- ChineseName-StudentID, e.g., 2014-11-24-张三-123456789.



```
*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.x
z=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.d
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*
.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.m
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;3
6:*.xspf=00;36:
MAIL=/var/mail/test
PATH=/home/test/bin:/home/test/.local/bin:/home/seed/bin:/usr/local/sb
in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-o
racle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/test/i_am_cool
QT_QPA_PLATFORMTHEME=appmenu-qt5
PWD=/home/wubinray/Desktop/lab1
JAVA_HOME=/usr/lib/jvm/java-8-oracle
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/test
LOGNAME=test
J2SDKDIR=/usr/lib/jvm/java-8-oracle
SSH_CONNECTION=127.0.0.1 34640 127.0.0.1 22
LESSOPEN=| /usr/bin/lesspipe %s
XDG_RUNTIME_DIR=/run/user/1002
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
_=./a.out
test@VM:/home/wubinray/Desktop/lab1$
```

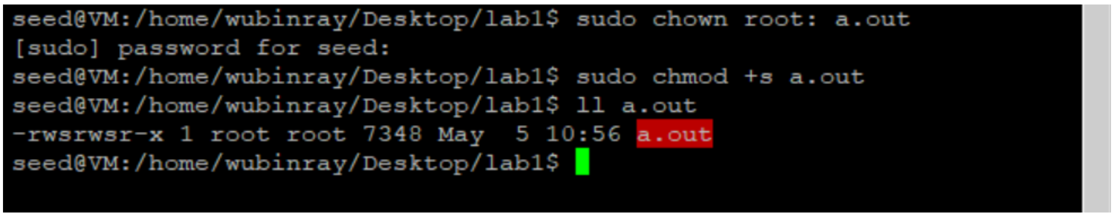
(六) The PATH Environment Variable and set-UID Programs :

(1) 写一个程序:



```
#include<stdlib.h>

int main(){
        system("ls");
        return 0;
}

~
```

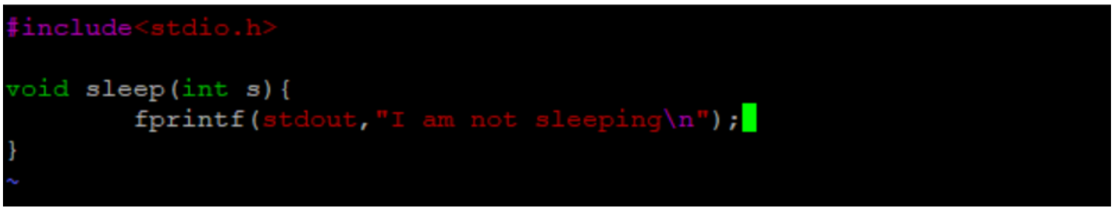
(2) 设定为 set-uid root program:



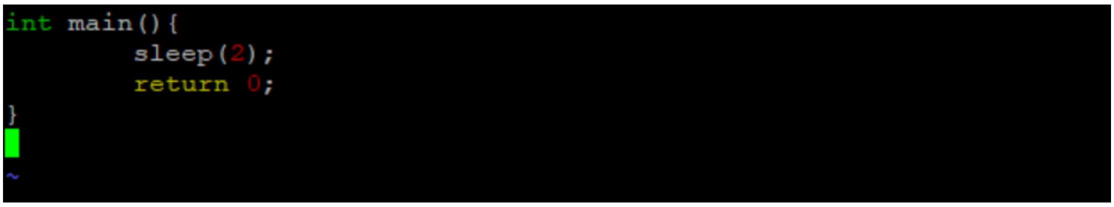
```
seed@VM:/home/wubinray/Desktop/lab1$ sudo chown root: a.out
[sudo] password for seed:
seed@VM:/home/wubinray/Desktop/lab1$ sudo chmod +s a.out
seed@VM:/home/wubinray/Desktop/lab1$ ll a.out
-rwsrwsr-x 1 root root 7348 May  5 10:56 a.out
seed@VM:/home/wubinray/Desktop/lab1$
```

(七) The LD_PRELOAD Environment Variable and Set-UID Programs :

(1) make mylib.c & myprog.c

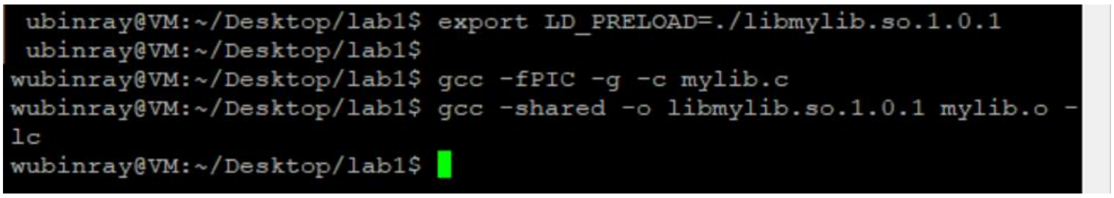


```
#include<stdio.h>

void sleep(int s){
        fprintf(stdout,"I am not sleeping\n");
}
~
```



```
int main(){
        sleep(2);
        return 0;
}

~
```

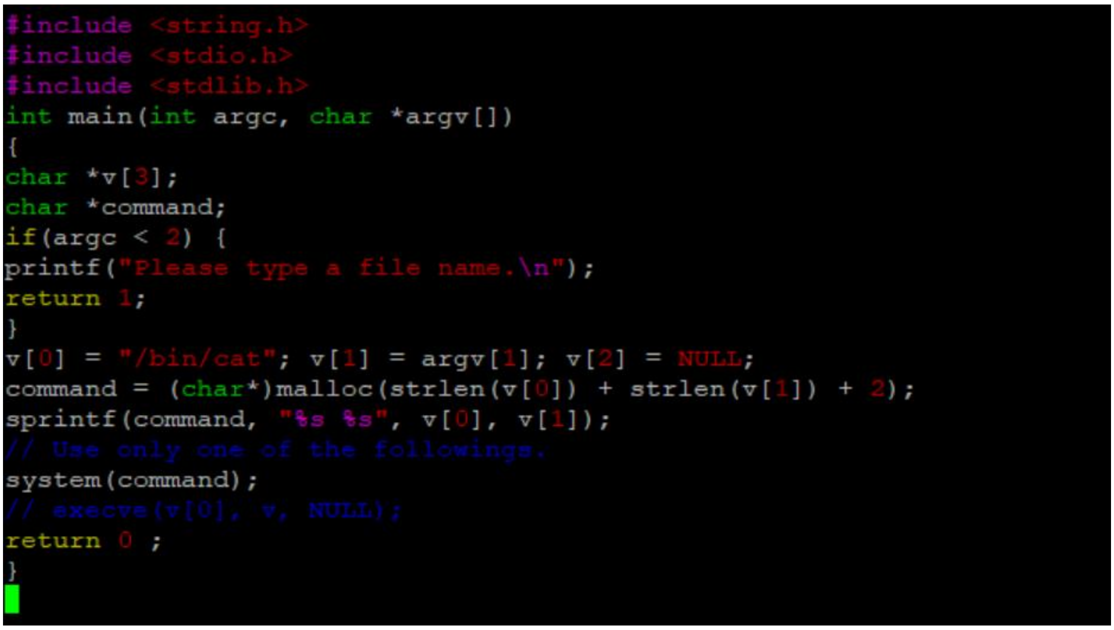
(2) export mylib to 环境变数 LD_RELOAD



```
 ubinray@VM:~/Desktop/lab1$ export LD_PRELOAD=./libmylib.so.1.0.1
 ubinray@VM:~/Desktop/lab1$
wubinray@VM:~/Desktop/lab1$ gcc -fPIC -g -c mylib.c
wubinray@VM:~/Desktop/lab1$ gcc -shared -o libmylib.so.1.0.1 mylib.o -
lc
wubinray@VM:~/Desktop/lab1$ 
```

(八) Invoking External Programs Using system() versus execve() :

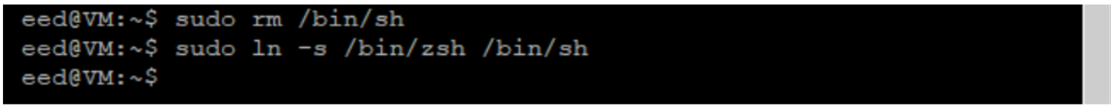
(1) program:



```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
char *v[3];
char *command;
if(argc < 2) {
printf("Please type a file name.\n");
return 1;
}
v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;
command = (char*)malloc(strlen(v[0]) + strlen(v[1]) + 2);
sprintf(command, "%s %s", v[0], v[1]);
// Use only one of the followings.
system(command);
// execve(v[0], v, NULL);
return 0 ;
}
```

(2) system()

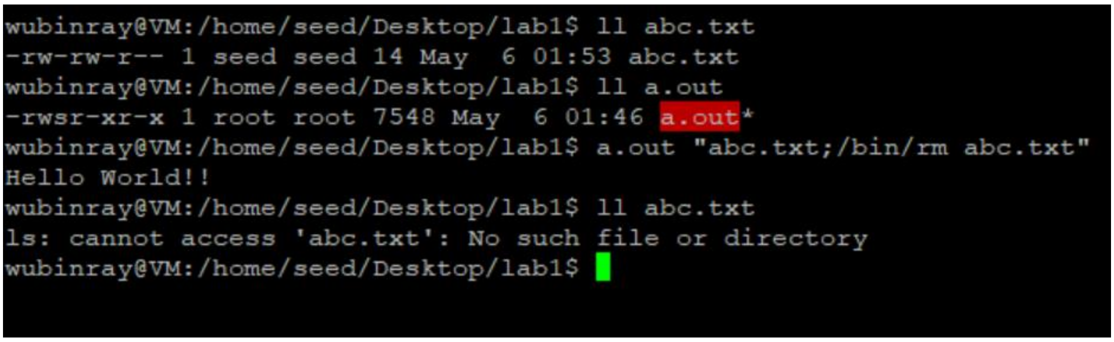
(a) 先把 ubuntu16 有自我保护的 sh 删除掉，换成没有保护的 zsh



```
eed@VM:~$ sudo rm /bin/sh
eed@VM:~$ sudo ln -s /bin/zsh /bin/sh
eed@VM:~$
```

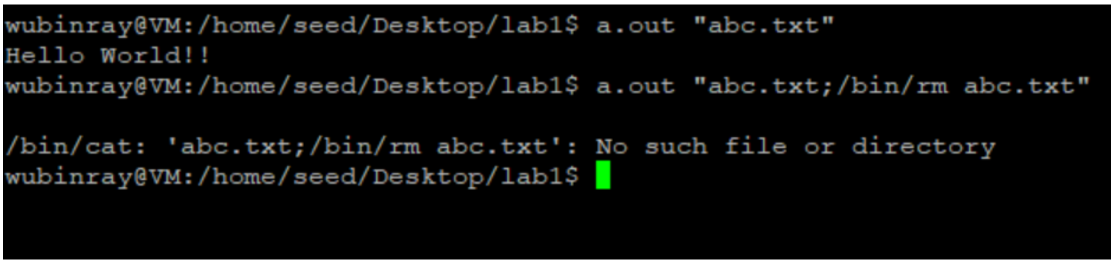
Report due Mon 11:59pm to MobileSecurity2014@163.com. You can write in either English or Chinese. Document naming convention: ExperimentDate- ChineseName-StudentID, e.g., 2014-11-24-张三-123456789.

(b) 执行删除挡案



```
wubinray@VM:/home/seed/Desktop/lab1$ ll abc.txt
-rw-rw-r-- 1 seed seed 14 May  6 01:53 abc.txt
wubinray@VM:/home/seed/Desktop/lab1$ ll a.out
-rwsr-xr-x 1 root root 7548 May  6 01:46 a.out*
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt;/bin/rm abc.txt"
Hello World!!
wubinray@VM:/home/seed/Desktop/lab1$ ll abc.txt
ls: cannot access 'abc.txt': No such file or directory
wubinray@VM:/home/seed/Desktop/lab1$
```

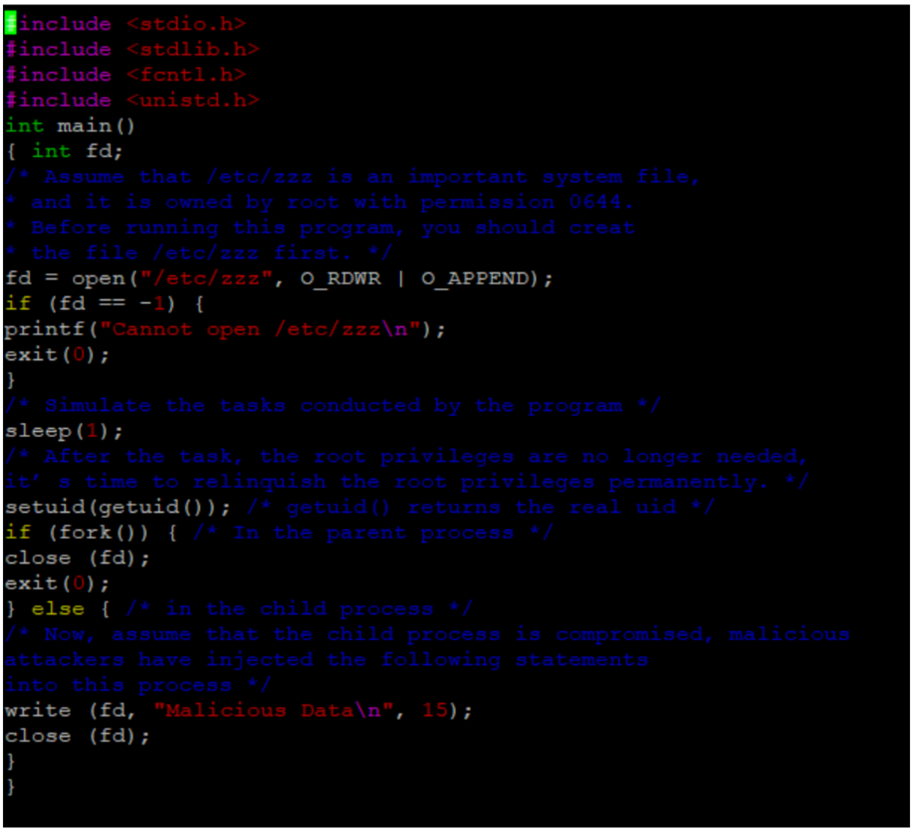
(3) execve():



```
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt"
Hello World!!
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt;/bin/rm abc.txt"

/bin/cat: 'abc.txt;/bin/rm abc.txt': No such file or directory
wubinray@VM:/home/seed/Desktop/lab1$
```

(九) Capability Leaking :

(1) 程序:

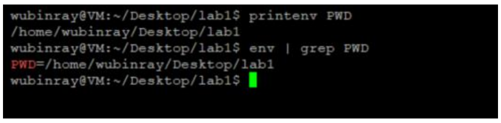


```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{ int fd;
/* Assume that /etc/zzz is an important system file,
* and it is owned by root with permission 0644.
* Before running this program, you should creat
* the file /etc/zzz first. */
fd = open("/etc/zzz", O_RDWR | O_APPEND);
if (fd == -1) {
printf("Cannot open /etc/zzz\n");
exit(0);
}
/* Simulate the tasks conducted by the program */
sleep(1);
/* After the task, the root privileges are no longer needed,
it’ s time to relinquish the root privileges permanently. */
setuid(getuid()); /* getuid() returns the real uid */
if (fork()) { /* In the parent process */
close (fd);
exit(0);
} else { /* in the child process */
/* Now, assume that the child process is compromised, malicious
attackers have injected the following statements
into this process */
write (fd, "Malicious Data\n", 15);
close (fd);
}
}
```

Report due Mon 11:59pm to MobileSecurity2014@163.com. You can write in either English or Chinese. Document naming convention: ExperimentDate- ChineseName-StudentID, e.g., 2014-11-24-张三-123456789.

三、Results and Analysis (结果与分析)

(一) Manipulate Enviroment Variables :



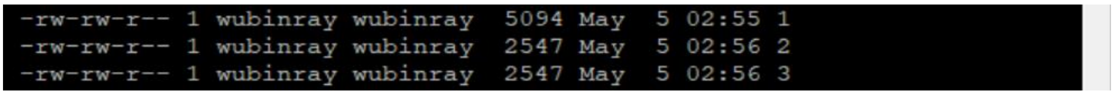
```
wubinray@VM:~/Desktop/lab1$ printenv PWD
/home/wubinray/Desktop/lab1
wubinray@VM:~/Desktop/lab1$ env | grep PWD
PWD=/home/wubinray/Desktop/lab1
wubinray@VM:~/Desktop/lab1$ 
```

(1) 两个结果都是 /home/wubinray/Desktop/lab1

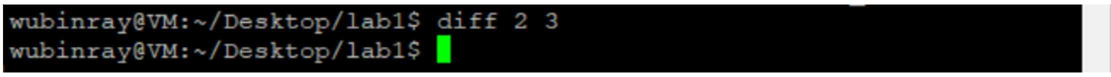
(2) env | grep PWD :

把 env 的 stdout 出来的内容 pipe 到 grep 程序去找”PWD”关键字

(二) Passing Environment Variables from Parent Process to Child Process :



```
-rw-rw-r-- 1 wubinray wubinray  5094 May  5 02:55 1
-rw-rw-r-- 1 wubinray wubinray  2547 May  5 02:56 2
-rw-rw-r-- 1 wubinray wubinray  2547 May  5 02:56 3
```



```
wubinray@VM:~/Desktop/lab1$ diff 2 3
wubinray@VM:~/Desktop/lab1$ 
```

(1) 1 : child 跟 parent 的 printenv()都正常执行

(2) 2 : 把 child 的 printenv()注解

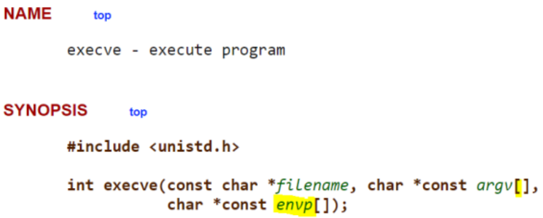
(3) 3 : 把 parent 的 printenv()注解

(4) 结论:

用 diff 确认 2 号跟 3 号挡案是一模一样的，因此 child process 会继承 parent process 的环境变数。

(三) Environment Variables and execve() :

(1) execve() 使用方法:



```
NAME       top

       execve - execute program

SYNOPSIS       top

       #include <unistd.h>

       int execve(const char *filename, char *const argv[],
                  char *const envp[]);
```

(2) 执行结果:

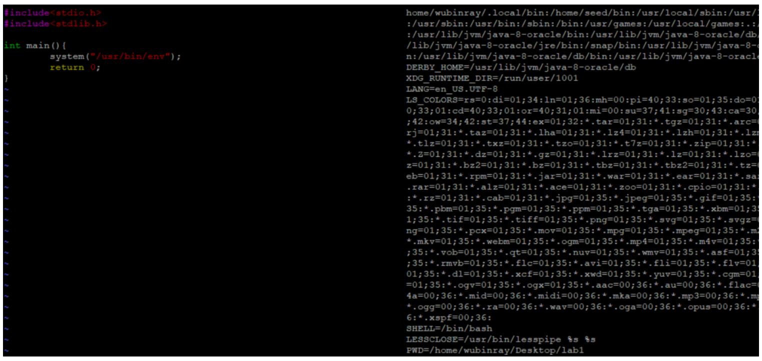
如果是 execve("/usr/bin/env", argv, NULL)，没有把环境变数传给新的 program，那么不会 print 出东西。

如果是 execve("/usr/bin/env", argv, environ)，有把环境变数传给新的 program，那么环境变数就会被 print 出来。

(3) 结论:

新的 program 不会继承旧的 program 的环境变数，除非有当做 argument 传给新的 program。

(四) Enviroment Variables and system() :



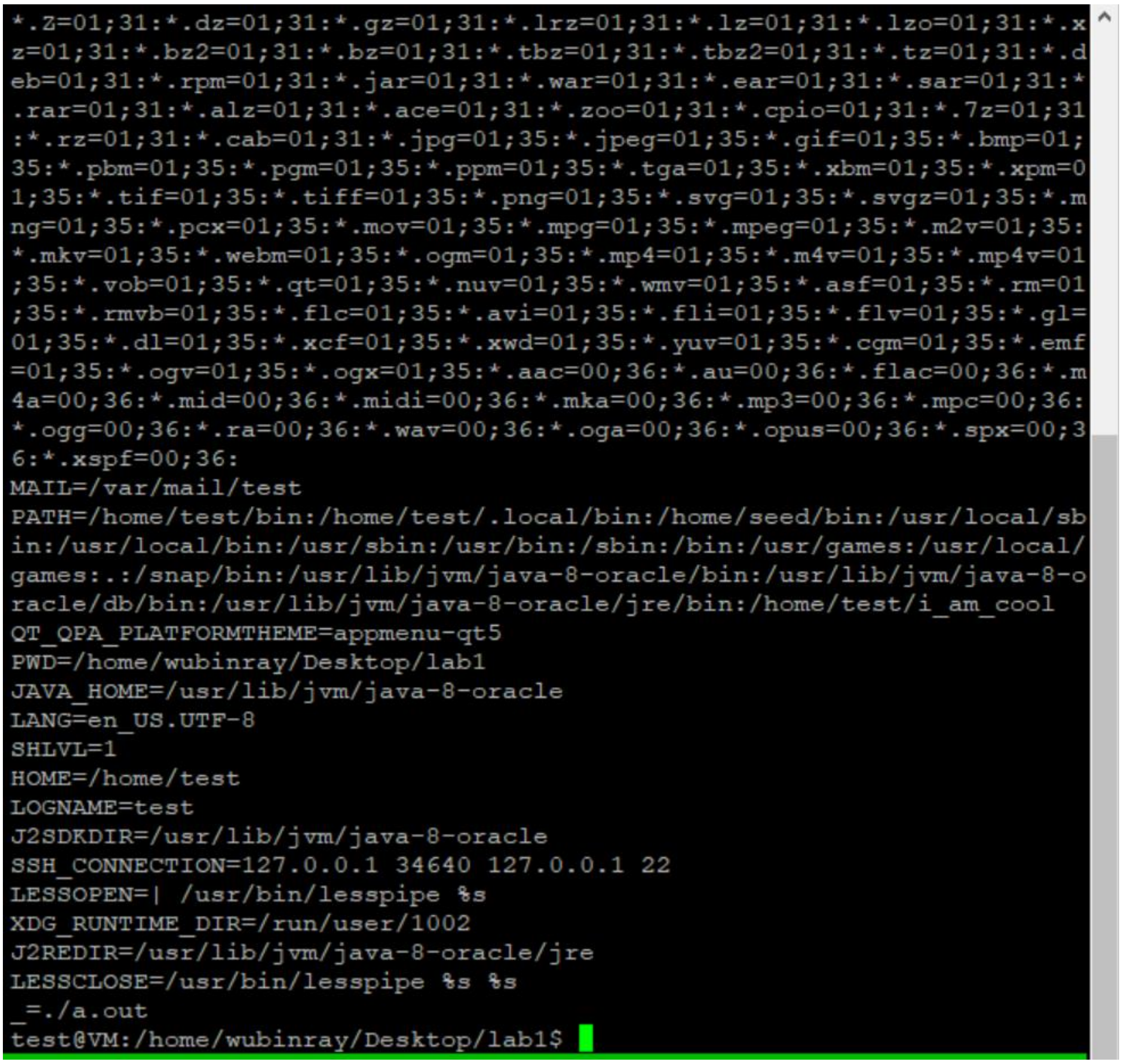
```
#include<stdio.h>
#include<stdlib.h>

int main(){
        system("/usr/bin/env");
        return 0;
}
```

```
home/wubinray/.local/bin:/home/seed/bin:/usr/local/sbin:/usr/
:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:
:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db
/lib/jvm/java-8-oracle/jre/bin:/snap/bin:/usr/lib/jvm/java-8-
n:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracl
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
XDG_RUNTIME_DIR=/run/user/1001
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=0
0;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30
;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=
rj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lz
*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*
*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=
z=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sa
.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;3
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;3
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.m
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*
6:*.xspf=00;36:
SHELL=/bin/bash
LESSCLOSE=/usr/bin/lesspipe %s %s
PWD=/home/wubinray/Desktop/lab1
```

环境变数有被 print 出来，因此是 execve()并且有把环境变数当做 argument 传下去的。

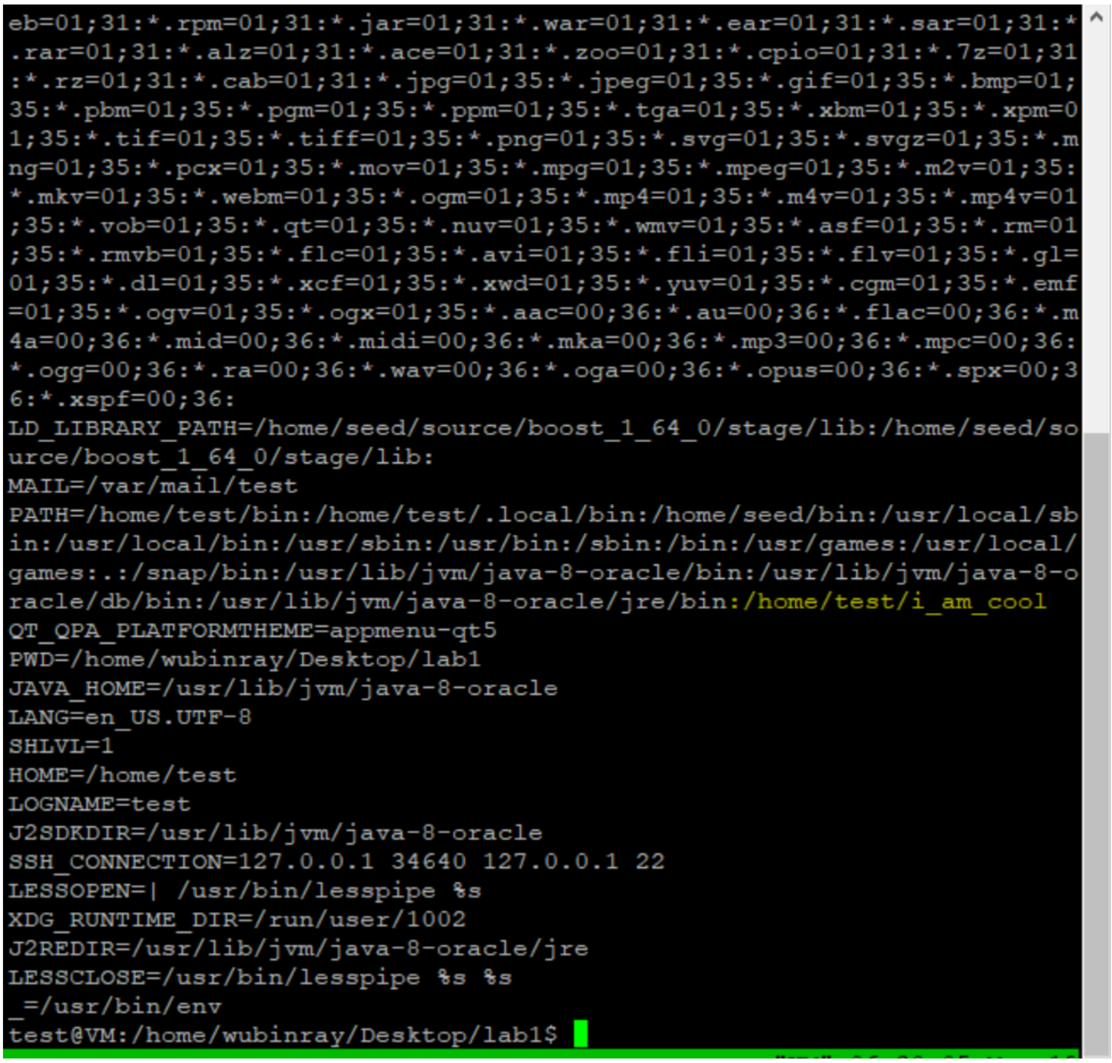
(五) Environment Variable and Set-UID Programs :

(1) 利用程序，发现环境变数没有被更改



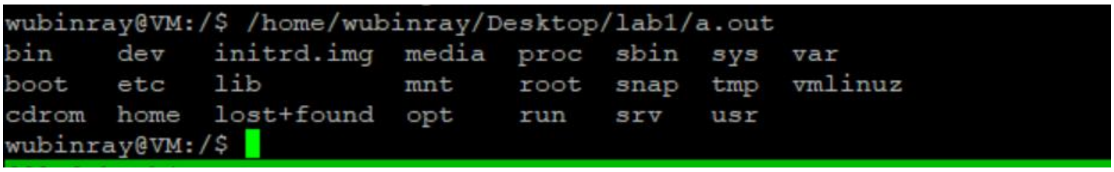
```
*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.x
z=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.d
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*
.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.m
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;3
6:*.xspf=00;36:
MAIL=/var/mail/test
PATH=/home/test/bin:/home/test/.local/bin:/home/seed/bin:/usr/local/sb
in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-o
racle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/test/i_am_cool
QT_QPA_PLATFORMTHEME=appmenu-qt5
PWD=/home/wubinray/Desktop/lab1
JAVA_HOME=/usr/lib/jvm/java-8-oracle
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/test
LOGNAME=test
J2SDKDIR=/usr/lib/jvm/java-8-oracle
SSH_CONNECTION=127.0.0.1 34640 127.0.0.1 22
LESSOPEN=| /usr/bin/lesspipe %s
XDG_RUNTIME_DIR=/run/user/1002
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
_=./a.out
test@VM:/home/wubinray/Desktop/lab1$
```

(2) env 指令，发现环境变数真的有被更改



```
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*
.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31
:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;
35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.m
ng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01
;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01
;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=
01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:
*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;3
6:*.xspf=00;36:
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/so
urce/boost_1_64_0/stage/lib:
MAIL=/var/mail/test
PATH=/home/test/bin:/home/test/.local/bin:/home/seed/bin:/usr/local/sb
in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-o
racle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/test/i_am_cool
QT_QPA_PLATFORMTHEME=appmenu-qt5
PWD=/home/wubinray/Desktop/lab1
JAVA_HOME=/usr/lib/jvm/java-8-oracle
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/test
LOGNAME=test
J2SDKDIR=/usr/lib/jvm/java-8-oracle
SSH_CONNECTION=127.0.0.1 34640 127.0.0.1 22
LESSOPEN=| /usr/bin/lesspipe %s
XDG_RUNTIME_DIR=/run/user/1002
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
_=/usr/bin/env
test@VM:/home/wubinray/Desktop/lab1$ 
```

(六) The PATH Environment Variable and Set-UID Programs：

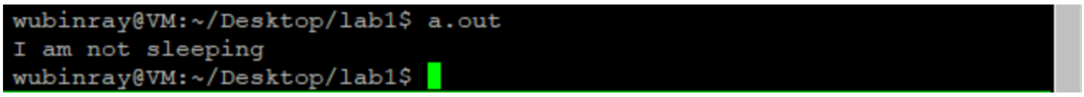


```
wubinray@VM:/$ /home/wubinray/Desktop/lab1/a.out
bin     dev   initrd.img  media  proc  sbin  sys  var
boot    etc   lib         mnt    root  snap  tmp  vmlinuz
cdrom   home  lost+found  opt    run   srv   usr
wubinray@VM:/$ 
```

Yes，这个程序是以 root 的特权在运行的，因为 owner 是 root，而且又有 set-uid。

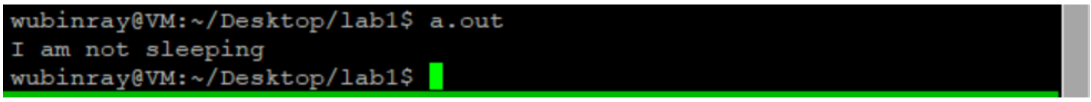
(七) The LD_PRELOAD Environment Variable and Set-UID Programs :

(1) Regular program ; Normal user:



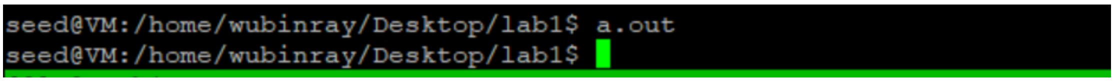
```
wubinray@VM:~/Desktop/lab1$ a.out
I am not sleeping
wubinray@VM:~/Desktop/lab1$ 
```

(2) Set-UID root program ; Normal user:



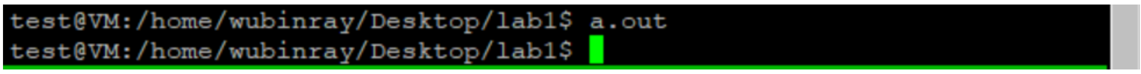
```
wubinray@VM:~/Desktop/lab1$ a.out
I am not sleeping
wubinray@VM:~/Desktop/lab1$ 
```

(3) Set-UID root program ; Root user:



```
seed@VM:/home/wubinray/Desktop/lab1$ a.out
seed@VM:/home/wubinray/Desktop/lab1$ 
```

(4) Set-UID user1 program ; user2 user:



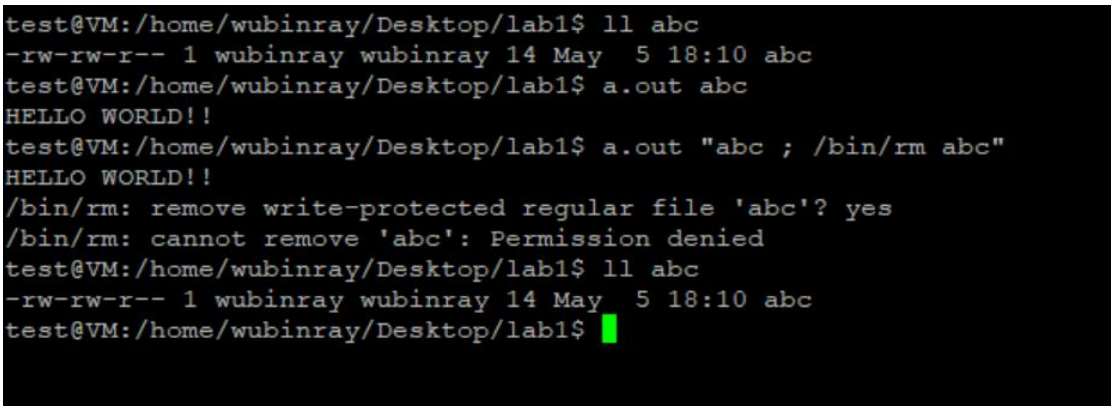
```
test@VM:/home/wubinray/Desktop/lab1$ a.out
test@VM:/home/wubinray/Desktop/lab1$ 
```

(5) 结论:

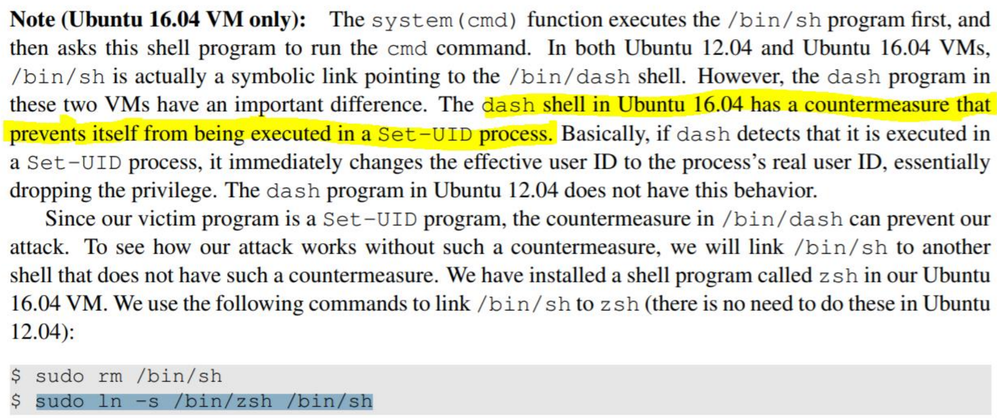
Child process 不会继承环境变数。

(八) Invoking External Programs Using system() and execve() :

(1) system() with 原生的 sh:



```
test@VM:/home/wubinray/Desktop/lab1$ ll abc
-rw-rw-r-- 1 wubinray wubinray 14 May  5 18:10 abc
test@VM:/home/wubinray/Desktop/lab1$ a.out abc
HELLO WORLD!!
test@VM:/home/wubinray/Desktop/lab1$ a.out "abc ; /bin/rm abc"
HELLO WORLD!!
/bin/rm: remove write-protected regular file 'abc'? yes
/bin/rm: cannot remove 'abc': Permission denied
test@VM:/home/wubinray/Desktop/lab1$ ll abc
-rw-rw-r-- 1 wubinray wubinray 14 May  5 18:10 abc
test@VM:/home/wubinray/Desktop/lab1$ 
```



Note (Ubuntu 16.04 VM only): The `system(cmd)` function executes the `/bin/sh` program first, and then asks this shell program to run the `cmd` command. In both Ubuntu 12.04 and Ubuntu 16.04 VMs, `/bin/sh` is actually a symbolic link pointing to the `/bin/dash` shell. However, the `dash` program in these two VMs have an important difference. The `dash` shell in Ubuntu 16.04 has a countermeasure that prevents itself from being executed in a `Set-UID` process. Basically, if `dash` detects that it is executed in a `Set-UID` process, it immediately changes the effective user ID to the process's real user ID, essentially dropping the privilege. The `dash` program in Ubuntu 12.04 does not have this behavior.

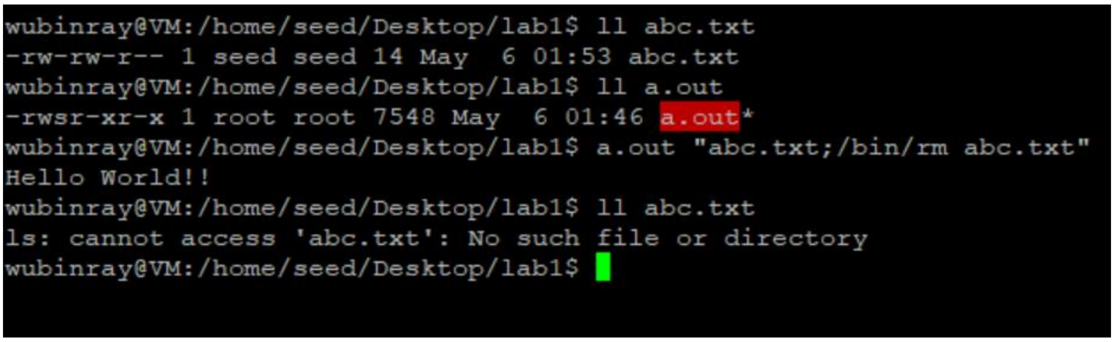
Since our victim program is a `Set-UID` program, the countermeasure in `/bin/dash` can prevent our attack. To see how our attack works without such a countermeasure, we will link `/bin/sh` to another shell that does not have such a countermeasure. We have installed a shell program called `zsh` in our Ubuntu 16.04 VM. We use the following commands to link `/bin/sh` to `zsh` (there is no need to do these in Ubuntu 12.04):

```
$ sudo rm /bin/sh
$ sudo ln -s /bin/zsh /bin/sh
```

在原生的 ubuntu16 不会有问题，因为系统有自我保护，可以保证系统的完整性。但是在 ubuntu12 不能保证。

Report due Mon 11:59pm to MobileSecurity2014@163.com. You can write in either English or Chinese. Document naming convention: ExperimentDate- ChineseName-StudentID, e.g., 2014-11-24-张三-123456789.

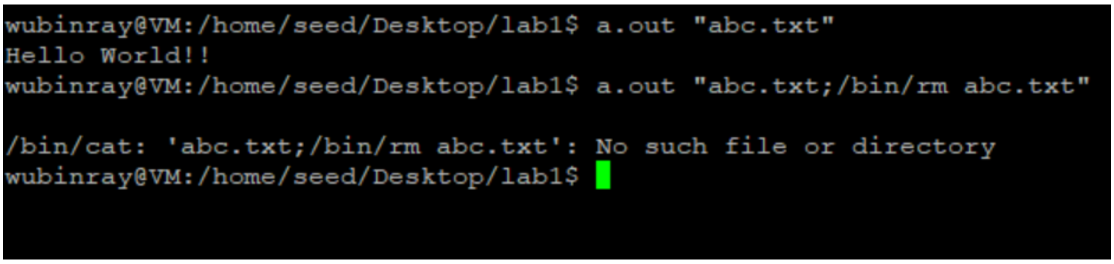
(2) system() with zsh 替换掉 sh (系统没有自我检测):



```
wubinray@VM:/home/seed/Desktop/lab1$ ll abc.txt
-rw-rw-r-- 1 seed seed 14 May  6 01:53 abc.txt
wubinray@VM:/home/seed/Desktop/lab1$ ll a.out
-rwsr-xr-x 1 root root 7548 May  6 01:46 a.out*
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt;/bin/rm abc.txt"
Hello World!!
wubinray@VM:/home/seed/Desktop/lab1$ ll abc.txt
ls: cannot access 'abc.txt': No such file or directory
wubinray@VM:/home/seed/Desktop/lab1$ 
```

挡案被删除掉了，这是一个漏洞。

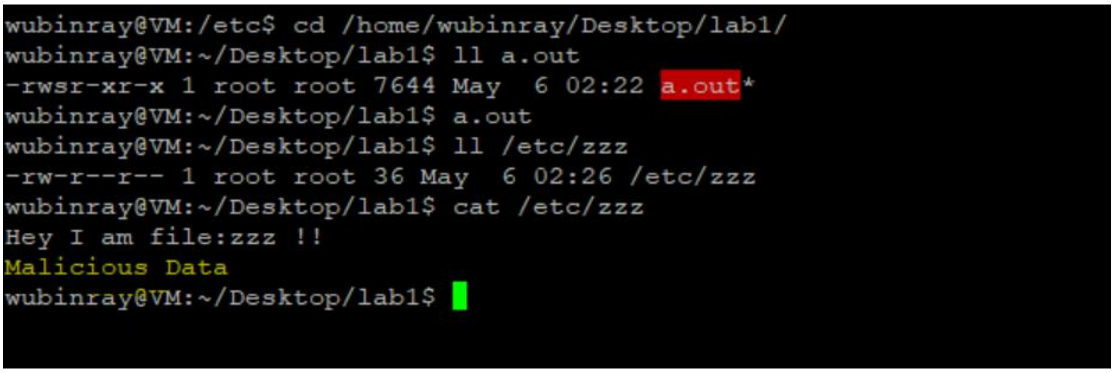
(3) execve():



```
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt"
Hello World!!
wubinray@VM:/home/seed/Desktop/lab1$ a.out "abc.txt;/bin/rm abc.txt"

/bin/cat: 'abc.txt;/bin/rm abc.txt': No such file or directory
wubinray@VM:/home/seed/Desktop/lab1$ 
```

挡案没有被删除掉，因为 execve()不是执行 shell，而是直接去执行 cat 这个程序，因此;后面的”/bin/rm abc.txt”都被当成 argument 丢到 cat 里头了。

(九) Capability Leaking :



```
wubinray@VM:/etc$ cd /home/wubinray/Desktop/lab1/
wubinray@VM:~/Desktop/lab1$ ll a.out
-rwsr-xr-x 1 root root 7644 May  6 02:22 a.out*
wubinray@VM:~/Desktop/lab1$ a.out
wubinray@VM:~/Desktop/lab1$ ll /etc/zzz
-rw-r--r-- 1 root root 36 May  6 02:26 /etc/zzz
wubinray@VM:~/Desktop/lab1$ cat /etc/zzz
Hey I am file:zzz !!
Malicious Data
wubinray@VM:~/Desktop/lab1$ 
```

挡案被修改了，代表 root 的特权还在我们程序的手上，还是可以执行特权的动作。