

Task 1: Manipulating Environment Variables

- Use `printenv` or `env` command to print out the environment variables. If you are interested in some particular environment variables, such as `PWD`, you can use "`printenv PWD`" or "`env | grep PWD`".

A screenshot of an Ubuntu desktop environment. On the left, a vertical dock contains icons for various applications: Dash (blue square), Nautilus (file folder), System Settings (gear), Software Center (red/green icon), and a terminal window (black background with white text). The terminal window shows a command-line session where the user checks the LANGUAGE environment variable, sets it to en_US, and then checks it again, resulting in an empty line. The desktop background is dark grey. The top right corner shows system status icons for network, battery, volume, and time (4:50 PM).

I used the grep command to search for the LANGUAGE environment variable from the list of environment variables.

- Use export and unset to set or unset environment variables. It should be noted that these two commands are not separate programs; they are two of the Bash's internal commands (you will not be able to find them outside of Bash).

```
[09/09/19]seed@VM:~$ env | grep LANGUAGE
Text Editor )]seed@VM:~$ export PATH=$PATH:/usr/local/bin
[09/09/19]seed@VM:~$ env | PATH
PATH: command not found
[09/09/19]seed@VM:~$ env| grep PATH
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:./home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin:/usr/local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
[09/09/19]seed@VM:~$ █
```

Here I unset the language variable using the unset command and set the Path variable(adding to the default PATH) using the export call.

Task 2:

Observations:

There are no differences between the environment variables of the child and the parent process.

This is clear from the output of the diff command showing that the child inherits all the environment variables from the parent process

```
[09/08/19]seed@VM:~$ gcc task2.c -o task2
[09/08/19]seed@VM:~$ task2 > child
[09/08/19]seed@VM:~$ gcc task2.c -o task2
task2.c: In function 'main':
task2.c:18:1: error: stray '\' in program
 \\\printenv();
^
task2.c:18:1: error: stray '\' in program
[09/08/19]seed@VM:~$ gcc task2.c -o task2
[09/08/19]seed@VM:~$ task2 > parent
[09/08/19]seed@VM:~$ diff
diff: missing operand after 'diff'
diff: Try 'diff --help' for more information.
[09/08/19]seed@VM:~$ diff parent child
[09/08/19]seed@VM:~$ diff child.txt parent.txt
```

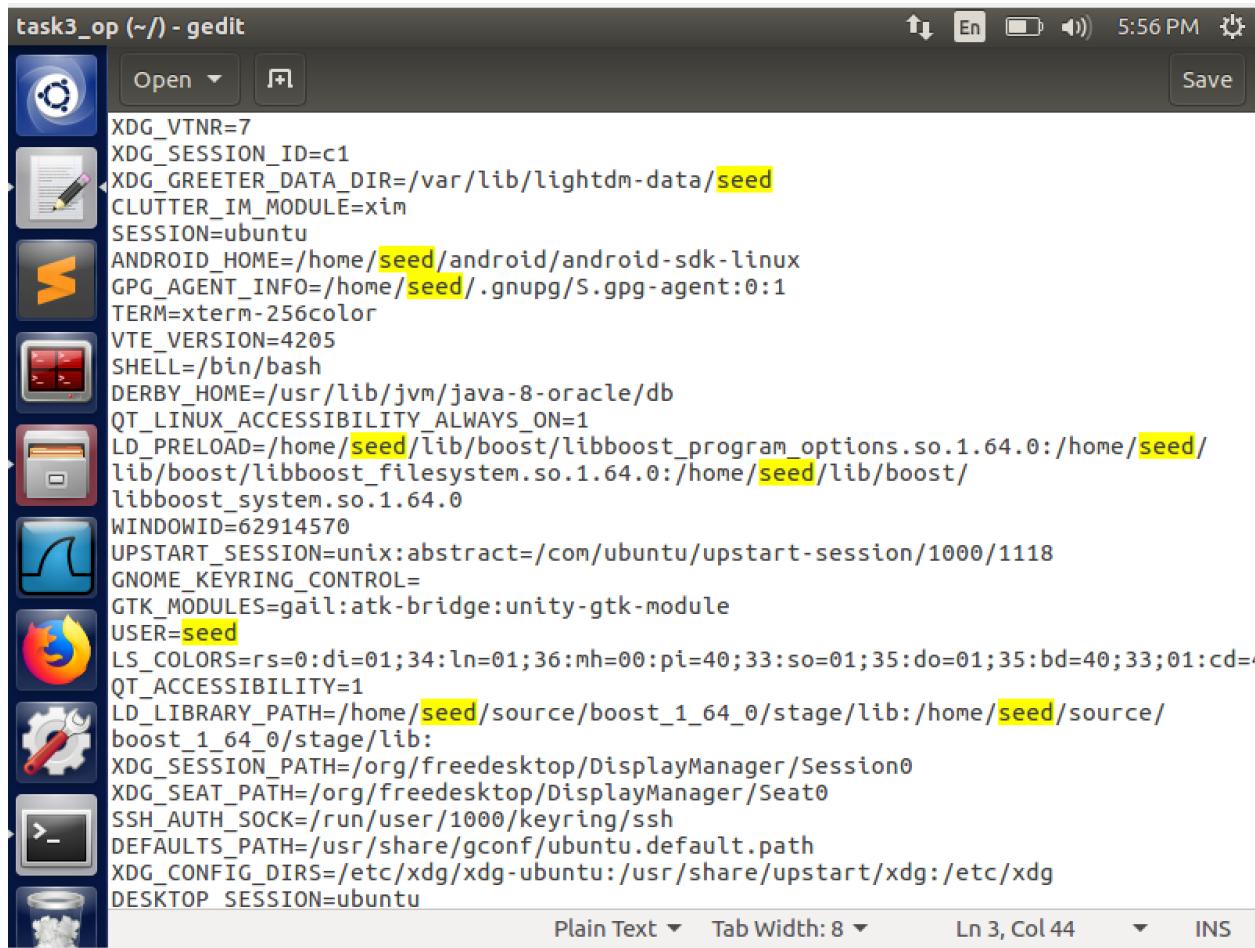
Task 3

Case 1

The execve call directly calls the OS to perform a given command. In the first case we are calling env using the argument 'argv' and not passing any environment variables. This renders a blank file when we print the environment variables using the env command, in the first scenario.

Case 2:

In case 2 we pass the array of environment variables 'environ' in the environment variables section of the execve call. This generates a file with the environment variables passed to it.



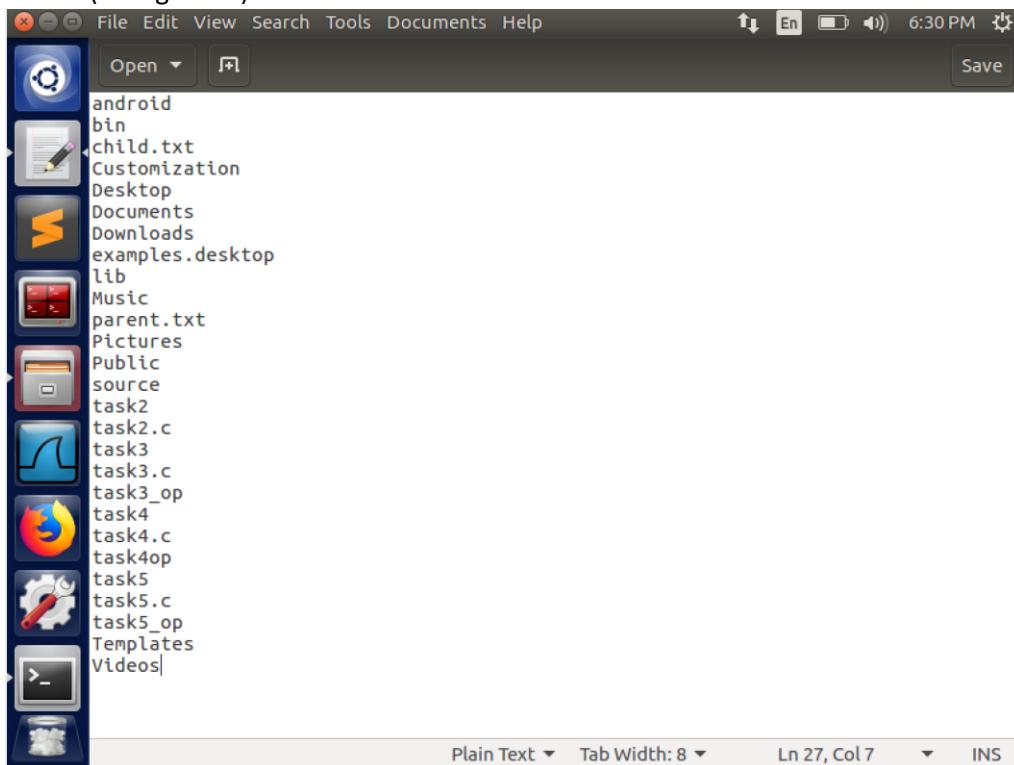
The screenshot shows a Gedit text editor window titled "task3_op (~/) - gedit". The window contains a large amount of text representing environment variables. Several lines of text are highlighted in yellow, including "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed", "LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0", and "XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0". The text is in plain text mode, and the status bar at the bottom indicates "Plain Text" and "Ln 3, Col 44".

```
XDG_VTNR=7
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=62914570
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1118
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=4
QT_ACCESSIBILITY=1
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
```

Task 4:

We use the `system()` function to run `/usr/bin/env`. We can verify from the screenshot that `system()` inherits the environment variables of the calling process. We can further verify this by changing '`/usr/bin/env`' to '`ls`'. This gives us the output as the list of all files in the default path

```
task4op (~/) - gedit
Open ▾  [+]
Save
DEFAULTS_PATH=/usr/share/gconf/ubuntu.defau
XDG_SESSION_ID=c1
TERM=xterm-256color
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK2_MODULES=overlay-scrollbar
PATH=/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:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
GDM_LANG=en_US
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_RUNTIME_DIR=/run/user/1000
COMPIZ_BIN_PATH=/usr/bin/
DISPLAY=:0
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=Unity
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=31;01:ex=01;32
XMODIFIERS=@im=ibus
XDG_SESSION_DESKTOP=ubuntu
XAUTHORITY=/home/seed/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
SHELL=/bin/bash
QT_ACCESSIBILITY=1
GDMSESSION=ubuntu
```

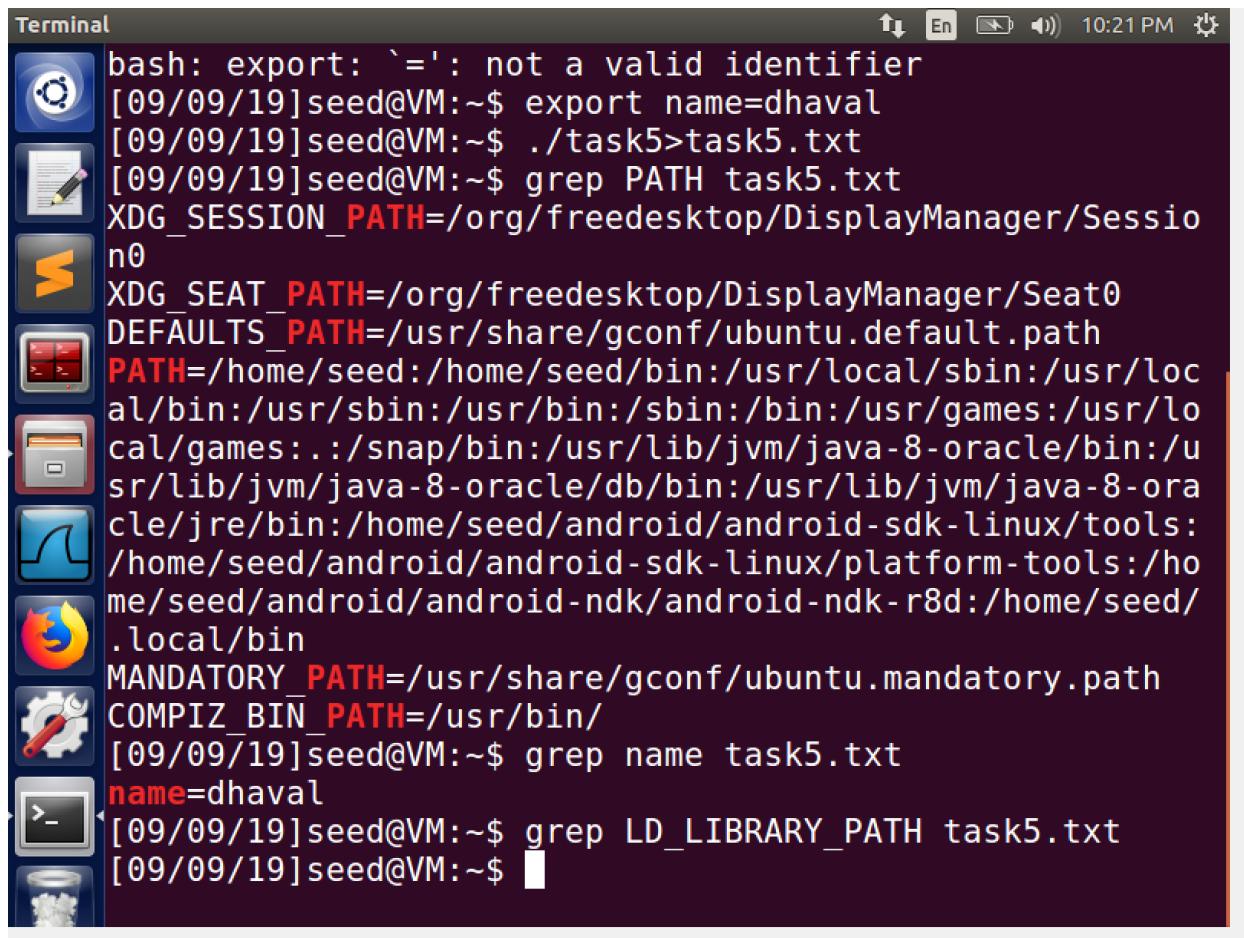
Case 2 (change to ls)**Task 5**

We compile the given program as task5 and make it a setuid program.

After that, we use the export command to set the environment variables to pass to the child process

We print all the environment variables in a txt file and notice that all the variables except the LD_LIBRARY_PATH variable is not inherited by the child process. This provides protection against malicious code.

```
[09/09/19]seed@VM:~$ gedit task5
[09/09/19]seed@VM:~$ gcc task5.c -o task5
[09/09/19]seed@VM:~$ sudo chown root task5
[sudo] password for seed:
[09/09/19]seed@VM:~$ sudo chmod 4755 task5
[09/09/19]seed@VM:~$ ll task5
-rwsr-xr-x 1 root seed 7396 Sep  9 22:14 task5
[09/09/19]seed@VM:~$ export PATH=/home/seed:$PATH
[09/09/19]seed@VM:~$ export LD_LIBRARY_PATH=test
[09/09/19]seed@VM:~$ export var = 123
bash: export: `=': not a valid identifier
bash: export: `123': not a valid identifier
[09/09/19]seed@VM:~$ export test = foo
bash: export: `=': not a valid identifier
[09/09/19]seed@VM:~$ export name=dhaval
[09/09/19]seed@VM:~$ ./task5>task5.txt
[09/09/19]seed@VM:~$ grep PATH task5.txt
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed:/home/seed/bin:/usr/local/sbin:/usr/loc
```



The screenshot shows a terminal window on an Ubuntu desktop. The terminal output is as follows:

```
Terminal
bash: export: `=': not a valid identifier
[09/09/19]seed@VM:~$ export name=dhaval
[09/09/19]seed@VM:~$ ./task5>task5.txt
[09/09/19]seed@VM:~$ grep PATH task5.txt
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed:/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:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
[09/09/19]seed@VM:~$ grep name task5.txt
name=dhaval
[09/09/19]seed@VM:~$ grep LD_LIBRARY_PATH task5.txt
[09/09/19]seed@VM:~$
```

Task 6

To run our command instead of the command defined we will use environment variables.

To do this we first change the environment variables. We append the environment variables with a path where we will store or malicious ls file which will get executed.

Export PATH=/home/seed:\$PATH does this

Now the system call will search in /home/seed before looking at /bin/ls

This is how we can run our code in a set uid program and gain root privilege to run our commands

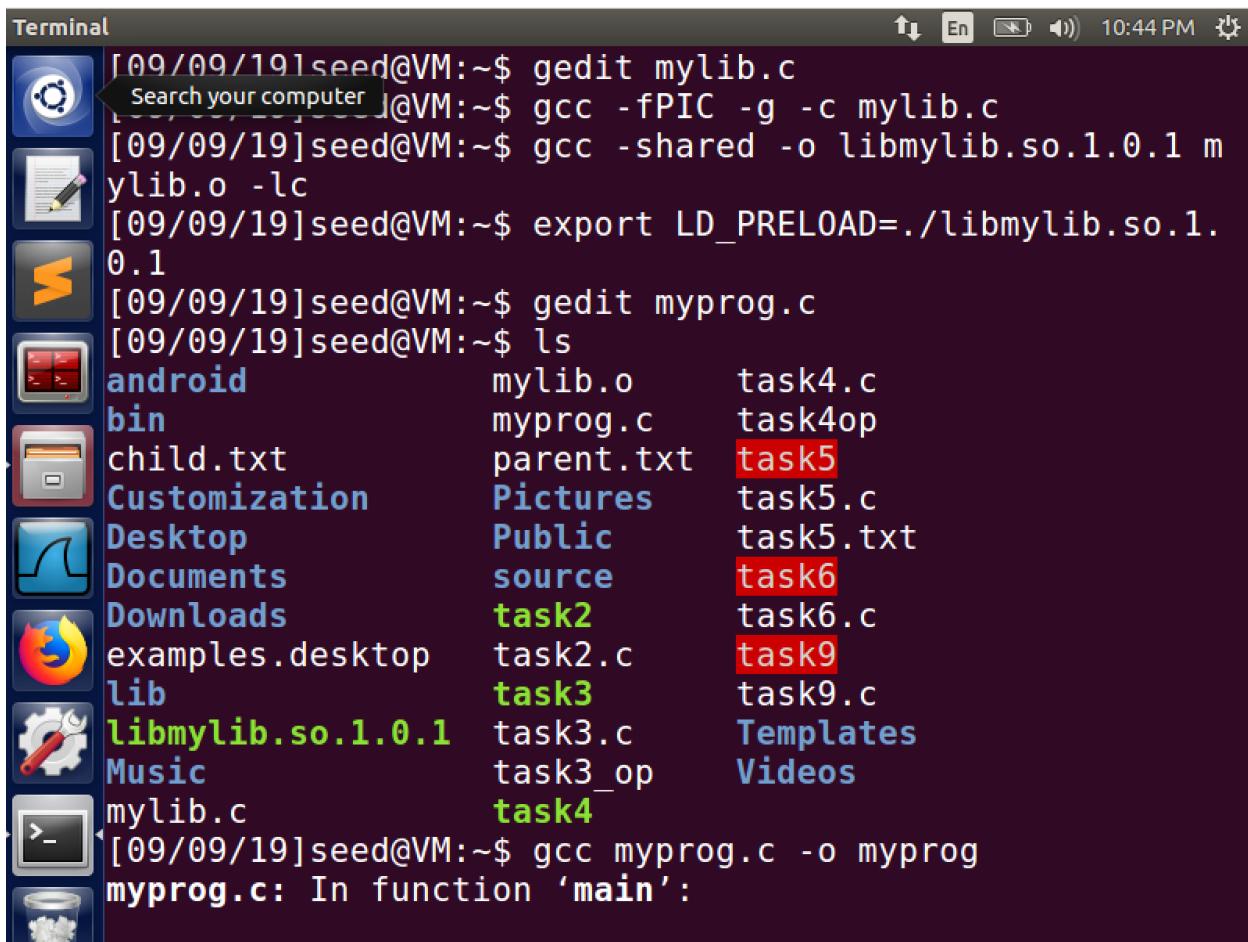
Terminal

```
[09/10/19]seed@VM:~$ gedit task6.c
[09/10/19]seed@VM:~$ gcc task6.c -o task6
task6.c: In function 'main':
task6.c:3:1: warning: implicit declaration of function
'system' [-Wimplicit-function-declaration]
system("ls");
^
[09/10/19]seed@VM:~$ sudo chown root task6
[sudo] password for seed:
[09/10/19]seed@VM:~$ sudo chmod 4755 task6
[09/10/19]seed@VM:~$ ll task6
-rwsr-xr-x 1 root seed 7348 Sep 10 04:08 task6
[09/10/19]seed@VM:~$ export PATH=/home/seed:$PATH
[09/10/19]seed@VM:~$ printenv | grep PATH
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed:/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lo
```

Terminal

```
[09/10/19]seed@VM:~$ export PATH=/home/seed:$PATH
[09/10/19]seed@VM:~$ printenv | grep PATH
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed:/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:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/
Firefox Web Browser
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
[09/10/19]seed@VM:~$ ./task6
hack
[09/10/19]seed@VM:~$
```

Task 7



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The terminal content shows the following steps:

```
[09/09/19]seed@VM:~$ gedit mylib.c
[09/09/19]seed@VM:~$ gcc -fPIC -g -c mylib.c
[09/09/19]seed@VM:~$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[09/09/19]seed@VM:~$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/09/19]seed@VM:~$ gedit myprog.c
[09/09/19]seed@VM:~$ ls
android           mylib.o      task4.c
bin               myprog.c    task4op
child.txt         parent.txt   task5
Customization     Pictures     task5.c
Desktop           Public       task5.txt
Documents         source      task6
Downloads         task2       task6.c
examples.desktop task2.c     task9
lib               task3       task9.c
libmylib.so.1.0.1 task3.c     Templates
Music             task3_op    Videos
mylib.c          task4
[09/09/19]seed@VM:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
```

In this task we compile a dll and change the path variable of the LOADER using the LD_LIBRARY.

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The terminal content is as follows:

```
[09/09/19]seed@VM:~$ ls
android           mylib.o      task4.c
bin               myprog.c    task4op
child.txt        parent.txt   task5
Customization    Pictures     task5.c
Desktop          Public       task5.txt
Documents         source      task6
Downloads         task2       task6.c
examples.desktop task2.c    task9
lib               task3       task9.c
libmylib.so.1.0.1 task3.c    Templates
Music             task3_op    Videos
mylib.c          task4

[09/09/19]seed@VM:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
myprog.c:3:1: warning: implicit declaration of function
'sleep' [-Wimplicit-function-declaration]
sleep(1);
^

[09/09/19]seed@VM:~$ ./myprog
sleeping!
```

We now run the program and we can see that the dll works with the output. After that we make the program to a setuid program with root privileges and now we pass our dll as a variable and we have successfully run our program in the setuid program. Now we try and run the same run program with another user Bob and try to export the same environment variable, the screenshot shows that we fail and our malicious sleep is not executed. This shows that the LD_PRELOAD variable has protection while executing with a different user.

```
root@VM: /home/seed          task4
mylib.c
[09/09/19]seed@VM:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
myprog.c:3:1: warning: implicit declaration of function
'sleep' [-Wimplicit-function-declaration]
sleep(1);
^
[09/09/19]seed@VM:~$ ./myprog
I am not sleeping!
[09/09/19]seed@VM:~$ sudo chown root myprog
[sudo] password for seed:
[09/09/19]seed@VM:~$ sudo chmod 4755 myprog
[09/09/19]seed@VM:~$ ll myprog
-rwsr-xr-x 1 root seed 7348 Sep  9 22:44 myprog
[09/09/19]seed@VM:~$ sudo su root
root@VM:/home/seed# export LD_PRELOAD=./libmylib.so.1.0
.1
system Settings
root@VM:/home/seed# printenv LD_PRELOAD
./libmylib.so.1.0.1
root@VM:/home/seed# ./myprog
I am not sleeping!
root@VM:/home/seed#
```

```

root@VM: /home/seed
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
root@VM:/home/seed# sudo chown bob myprog
root@VM:/home/seed# sudo chmod 4755 myprog
root@VM:/home/seed# ll myprog
-rwsr-xr-x 1 bob seed 7348 Sep  9 22:44 myprog*
root@VM:/home/seed# su seed
[09/09/19]seed@VM:~$ export LD_PRELOAD=./libmylib.so.1.
0.1
[09/09/19]seed@VM:~$ ./myprog
[09/09/19]seed@VM:~$ 

```

Task 8

We first define the `system()` and `execve()` programs to make the distinction between the cases clear.

`System()`

Uses the program owners shell and passes all the parameters as is. Hence there is a possibility of malicious execution.

`Execve(function to call,parameters to pass,environment variables)`

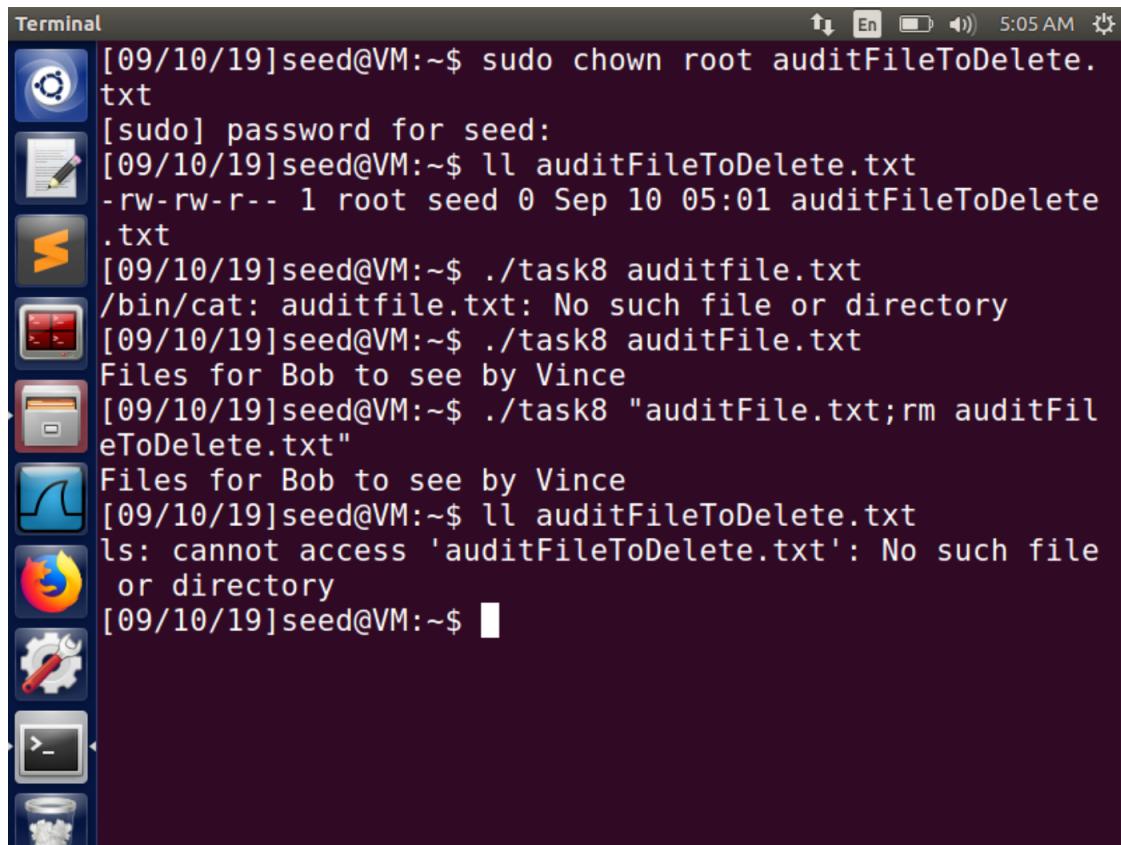
The function declaration of the `execve` shows why it is much better than `System()`

Scenario:

We create two files; namely `auditFileToDelete.txt` and `auditFile.txt`, both of them are owned by root and hence Bob has permission only to use them.

Case 1: `System()`

We separate our malicious code to delete the audit file using a semicolon the `System()` function takes it as is and deletes the file. Screenshot 1 proves this.

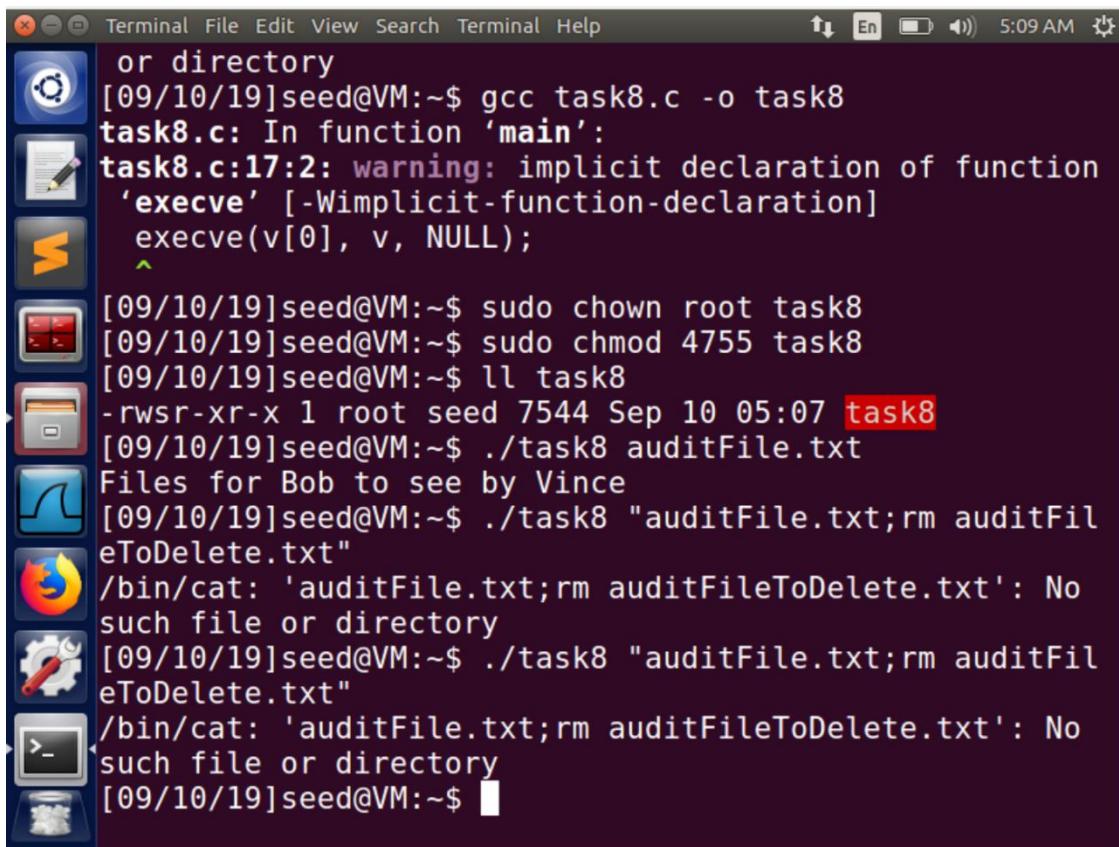


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The terminal content is as follows:

```
[09/10/19]seed@VM:~$ sudo chown root auditFileToDelete.txt
[sudo] password for seed:
[09/10/19]seed@VM:~$ ll auditFileToDelete.txt
-rw-rw-r-- 1 root seed 0 Sep 10 05:01 auditFileToDelete.txt
[09/10/19]seed@VM:~$ ./task8 auditfile.txt
/bin/cat: auditfile.txt: No such file or directory
[09/10/19]seed@VM:~$ ./task8 auditFile.txt
Files for Bob to see by Vince
[09/10/19]seed@VM:~$ ./task8 "auditFile.txt;rm auditFileToDelete.txt"
Files for Bob to see by Vince
[09/10/19]seed@VM:~$ ll auditFileToDelete.txt
ls: cannot access 'auditFileToDelete.txt': No such file or directory
[09/10/19]seed@VM:~$
```

Case 2 Execve()

Doing the same attack with execve() is not possible as the function to call is defined inside the program Vince created. All our arguments are actually passed as arguments itself making it tougher to attack or delete the file.



The screenshot shows a terminal window on an Ubuntu desktop. The terminal output is as follows:

```
or directory
[09/10/19]seed@VM:~$ gcc task8.c -o task8
task8.c: In function 'main':
task8.c:17:2: warning: implicit declaration of function
  'execve' [-Wimplicit-function-declaration]
    execve(v[0], v, NULL);
^
[09/10/19]seed@VM:~$ sudo chown root task8
[09/10/19]seed@VM:~$ sudo chmod 4755 task8
[09/10/19]seed@VM:~$ ll task8
-rwsr-xr-x 1 root seed 7544 Sep 10 05:07 task8
[09/10/19]seed@VM:~$ ./task8 auditFile.txt
Files for Bob to see by Vince
[09/10/19]seed@VM:~$ ./task8 "auditFile.txt;rm auditFil
eToDelete.txt"
/bin/cat: 'auditFile.txt;rm auditFileToDelete.txt': No
such file or directory
[09/10/19]seed@VM:~$ ./task8 "auditFile.txt;rm auditFil
eToDelete.txt"
/bin/cat: 'auditFile.txt;rm auditFileToDelete.txt': No
such file or directory
[09/10/19]seed@VM:~$
```

Task 9

Screenshot 1

Shows the compilation of the task9 program and the creation of binary file task9.

Screenshot 2

Here we make the task9 program a SetUID program using the chown and chmod commands and setting the setuid bit using the su privileges which makes the effective user id as root.

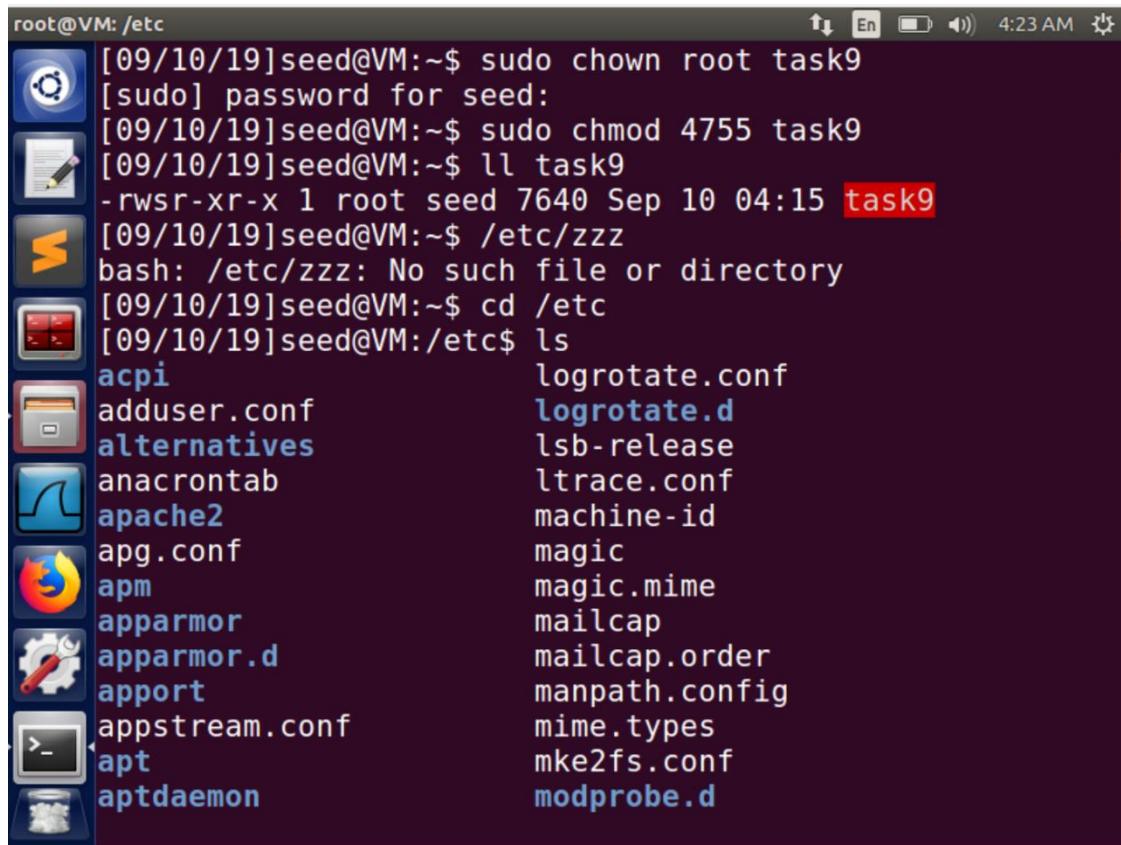
Screenshot 3

Shows creation of /etc/zzz file we change the rwx privileges so exemplify that only root can edit it.

Screenshot 4

We run task9, which runs the malicious code showing that the capability to write the file has been leaked by the file writer variable. The solution would be to get rid of such a file opener as soon as we finish using it.

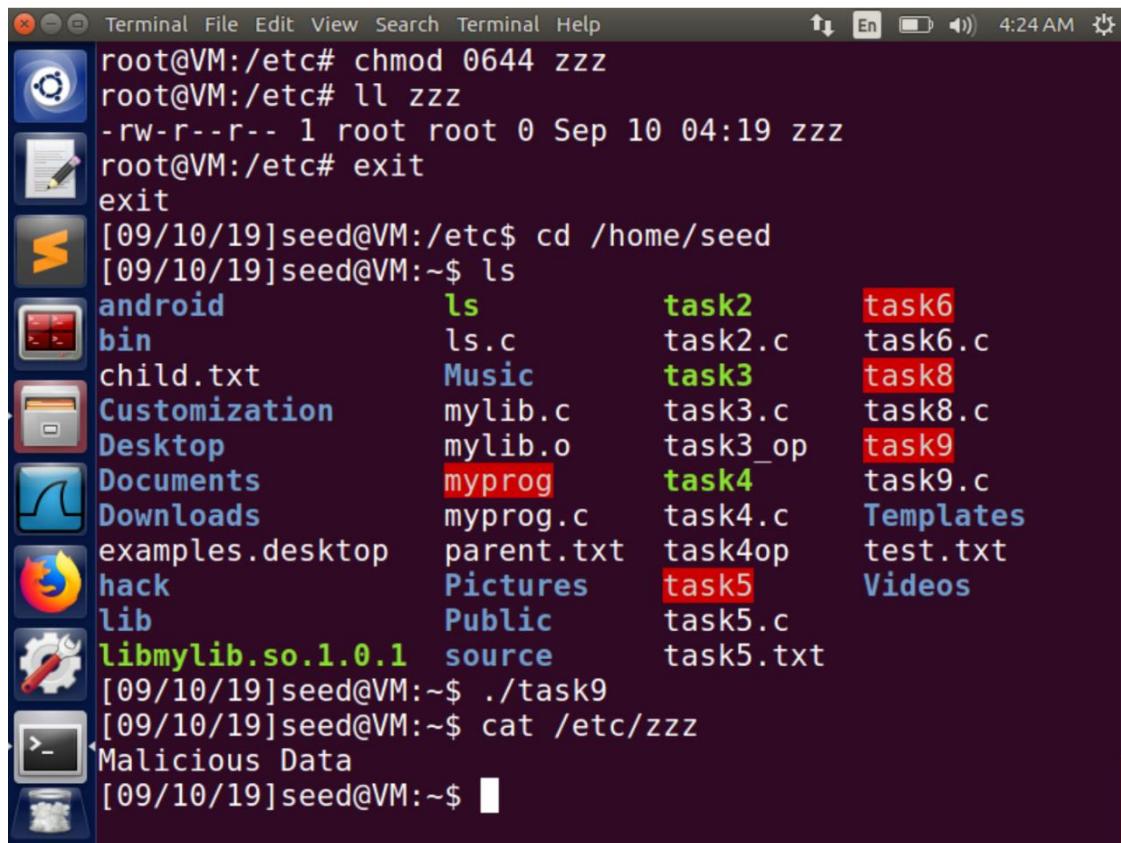
```
root@VM:/etc [09/10/19]seed@VM:~$ gedit task9.c
[09/10/19]seed@VM:~$ gcc task9.c -o task9
task9.c: In function 'main':
task9.c:16:1: warning: implicit declaration of function
'sleep' [-Wimplicit-function-declaration]
sleep(1);
^
task9.c:19:1: warning: implicit declaration of function
'setuid' [-Wimplicit-function-declaration]
setuid(getuid()); /* getuid() returns the real uid */
^
task9.c:19:8: warning: implicit declaration of function
'getuid' [-Wimplicit-function-declaration]
setuid(getuid()); /* getuid() returns the real uid */
^
task9.c:20:5: warning: implicit declaration of function
'fork' [-Wimplicit-function-declaration]
if (fork()) { /* In the parent process */
^
task9.c:21:1: warning: implicit declaration of function
'close' [-Wimplicit-function-declaration]
close (fd);
```



The screenshot shows a Linux desktop environment, likely Ubuntu, with a terminal window open. The terminal window has a dark background and displays a root shell session. The session starts with the command `sudo chown root task9`, followed by a password prompt. Then, `chmod 4755 task9` is run, making the file executable. The command `ll task9` is used to list the file's permissions, showing it has execute permission for root. A subsequent attempt to run `/etc/zzz` fails with the message "No such file or directory". The user then changes to the /etc directory and lists its contents, which include various configuration files like logrotate.conf, lsb-release, and mailcap.order.

```
[09/10/19]seed@VM:~$ sudo chown root task9
[sudo] password for seed:
[09/10/19]seed@VM:~$ sudo chmod 4755 task9
[09/10/19]seed@VM:~$ ll task9
-rwsr-xr-x 1 root seed 7640 Sep 10 04:15 task9
[09/10/19]seed@VM:~$ /etc/zzz
bash: /etc/zzz: No such file or directory
[09/10/19]seed@VM:~$ cd /etc
[09/10/19]seed@VM:/etc$ ls
acpi                  logrotate.conf
adduser.conf          logrotate.d
alternatives          lsb-release
anacrontab            ltrace.conf
apache2                machine-id
apg.conf               magic
apm                   magic.mime
apparmor              mailcap
apparmor.d             mailcap.order
apport                 manpath.config
appstream.conf         mime.types
apt                   mke2fs.conf
aptdaemon             modprobe.d
```

```
root@VM:/etc [09/10/19]seed@VM:/etc$ gedit zzz
[09/10/19]seed@VM:/etc$ sudo su root
root@VM:/etc# touch zzz
root@VM:/etc# chmod 0644 zzz
root@VM:/etc# ll zzz
-rw-r--r-- 1 root root 0 Sep 10 04:19 zzz
root@VM:/etc# exit
exit
[09/10/19]seed@VM:/etc$ cd /home/seed
[09/10/19]seed@VM:~/ls
android          ls        task2      task6
bin              ls.c      task2.c    task6.c
child.txt        Music     task3      task8
Customization   mylib.c   task3.c    task8.c
Desktop          mylib.o   task3_op   task9
Documents        myprog    task4      task9.c
Downloads        myprog.c  task4.c    Templates
examples.desktop parent.txt task4op    test.txt
hack             Pictures   task5      Videos
lib              Public    task5.c    task5.txt
libmylib.so.1.0.1 source    source
[09/10/19]seed@VM:~/ls ./task9
```



Terminal File Edit View Search Terminal Help

root@VM:/etc# chmod 0644 zzz
root@VM:/etc# ll zzz
-rw-r--r-- 1 root root 0 Sep 10 04:19 zzz
root@VM:/etc# exit
exit

[09/10/19]seed@VM:/etc\$ cd /home/seed

[09/10/19]seed@VM:~\$ ls

android	ls	task2	task6
bin	ls.c	task2.c	task6.c
child.txt	Music	task3	task8
Customization	mylib.c	task3.c	task8.c
Desktop	mylib.o	task3_op	task9
Documents	myprog	task4	task9.c
Downloads	myprog.c	task4.c	Templates
examples.desktop	parent.txt	task4op	test.txt
hack	Pictures	task5	Videos
lib	Public	task5.c	
libmylib.so.1.0.1	source	task5.txt	

[09/10/19]seed@VM:~\$./task9

[09/10/19]seed@VM:~\$ cat /etc/zzz

Malicious Data

[09/10/19]seed@VM:~\$