

Environment Variable and Set-UID Lab

Clayton B. Hodges

Colorado Mesa University

CSCI 420: Software Security

Contents

1.	Manipulating environment variables.....	3
1.1	Screenshots and Summaries	3
2.	Inheriting environment variables from parents.....	4
2.1	Screenshots and Summaries	4
3.	Environment variables and execve().....	5
3.1	Screenshot and Summary	5
4.	Environment variables and system()	6
4.1	Screenshot and Summary	6
5.	Environment variable and Set-UID Programs	6

5.1	Screenshots and Summaries	6
6.	The PATH Environment variable and Set-UID Programs	7
6.1	Screenshot and Summary	7
7.	The LD_PRELOAD environment variable and Set-UID Programs.....	8
7.1	Screenshots and Summaries	8
8.	Invoking external program using system() versus execve().....	9
8.1	Summary	9
9.	Capability Leaking	9
9.1	Screenshot and Summary	9

1. Manipulating environment variables

1.1 Screenshots and Summaries

```
[02/03/24]seed@VM: ~/.../Labsetup$ env
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1891,unix/VM:/tmp/.ICE-unix/1891
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=1841
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Downloads/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
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
IM_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;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.a
rj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:
```

I simply printed out the environment variables using the 'env' command.

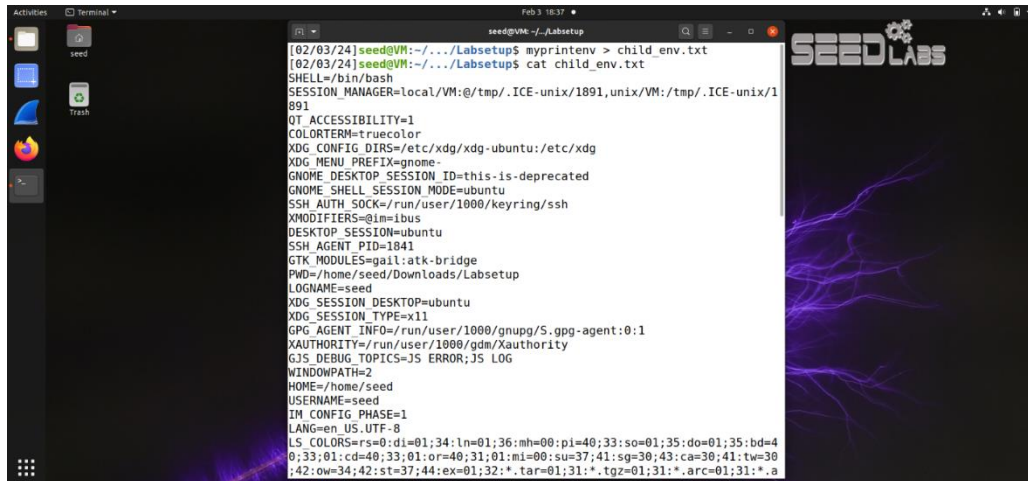
```
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/c571f7c4_935e_4ba7_ae
b5_22daed906014
INVOCATION_ID=702ad257f9ed4dc93bd90126fa62111
MANAGERPID=1633
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.69
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=8:35133
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib
/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr
/games:/usr/local/games:/snap/bin:
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
OLDPWD=/home/seed/Downloads
~/usr/bin/env
[02/03/24]seed@VM: ~/.../Labsetup$ export TEST="This is a test value."
[02/03/24]seed@VM: ~/.../Labsetup$ echo $TEST
This is a test value.
[02/03/24]seed@VM: ~/.../Labsetup$ unset TEST
[02/03/24]seed@VM: ~/.../Labsetup$ echo $TEST

[02/03/24]seed@VM: ~/.../Labsetup$
```

I created and destroyed an environment variable designated 'TEST' using the 'export' and 'unset' commands respectively. I also printed out these changes to the environment variable using the 'echo' command.

2. Inheriting environment variables from parents

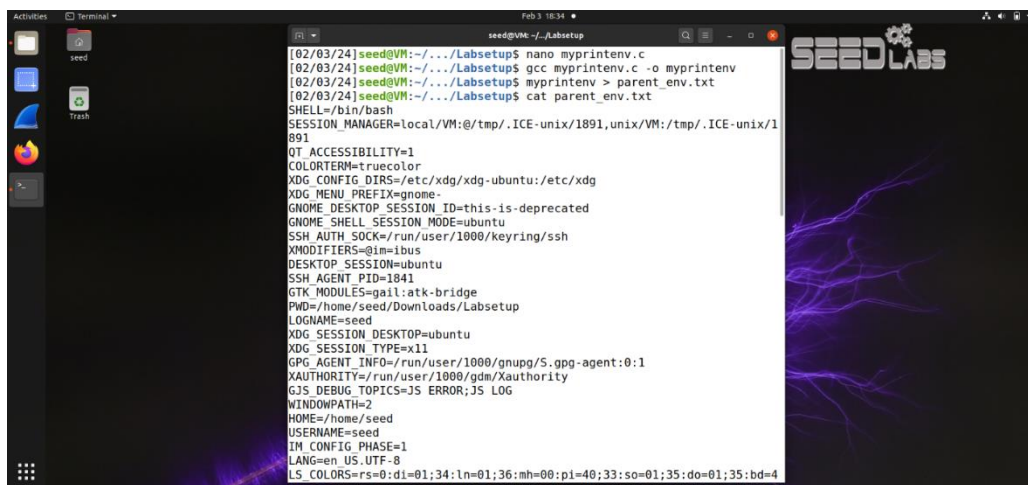
2.1 Screenshots and Summaries



```

[02/03/24]seed@VM:~/../Labsetup$ myprintenv > child_env.txt
[02/03/24]seed@VM:~/../Labsetup$ cat child_env.txt
SHELL=/bin/bash
SESSION_MANAGER=local/VM:/tmp/.ICE-unix/1891,unix/VM:/tmp/.ICE-unix/1891
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=1841
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Downloads/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
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
IM_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;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.a
  
```

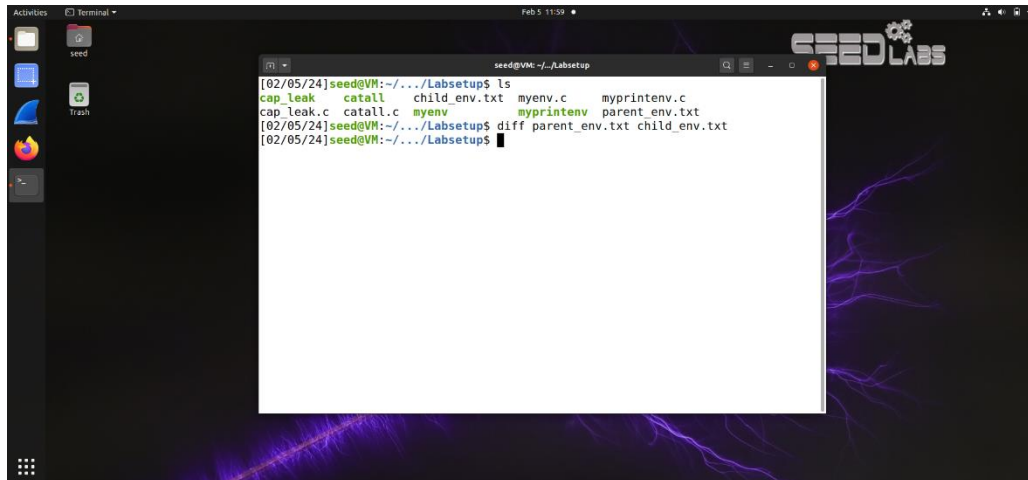
I compiled and ran the provided 'myprintenv' program and redirected its output to a text file.



```

[02/03/24]seed@VM:~/../Labsetup$ nano myprintenv.c
[02/03/24]seed@VM:~/../Labsetup$ gcc myprintenv.c -o myprintenv
[02/03/24]seed@VM:~/../Labsetup$ myprintenv > parent_env.txt
[02/03/24]seed@VM:~/../Labsetup$ cat parent_env.txt
SHELL=/bin/bash
SESSION_MANAGER=local/VM:/tmp/.ICE-unix/1891,unix/VM:/tmp/.ICE-unix/1891
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=1841
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Downloads/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
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
IM_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;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.a
  
```

I modified the provided program to print parent process results, noticing no differences in its output text file.



```

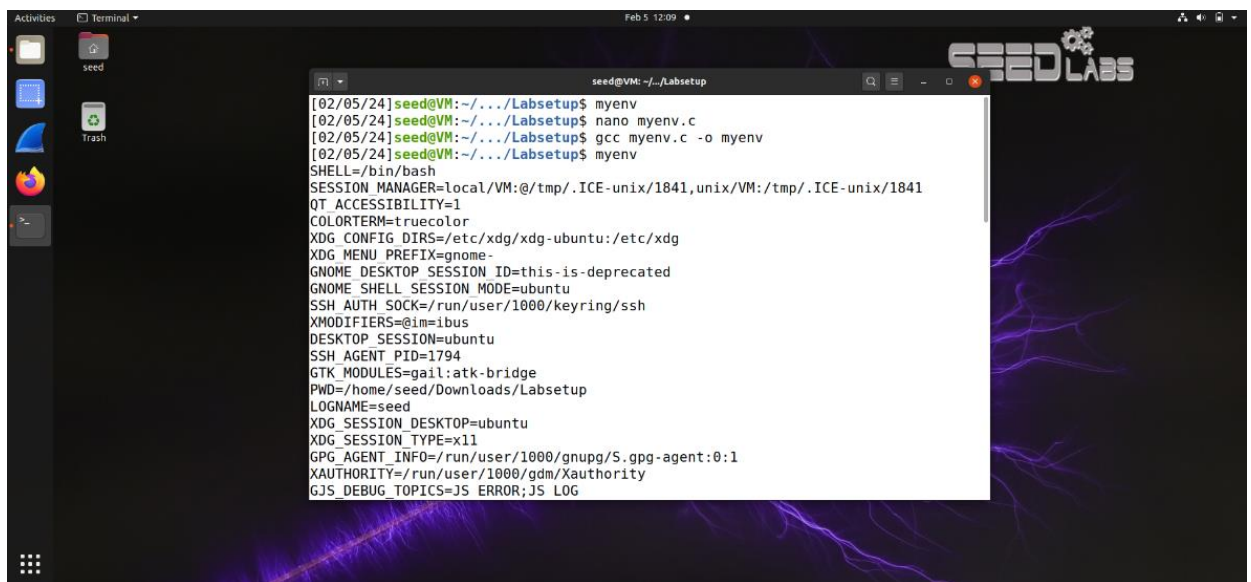
[02/05/24]seed@VM:~/../Labsetup$ ls
cap_leak  catall  child_env.txt  myenv.c  myprintenv.c
cap_leak.c  catall.c  myenv  myprintenv  parent_env.txt
[02/05/24]seed@VM:~/../Labsetup$ diff parent_env.txt child_env.txt
[02/05/24]seed@VM:~/../Labsetup$

```

I used the 'diff' command to check for any differences between the two output text files, there were none.

3. Environment variables and execve()

3.1 Screenshot and Summary



```

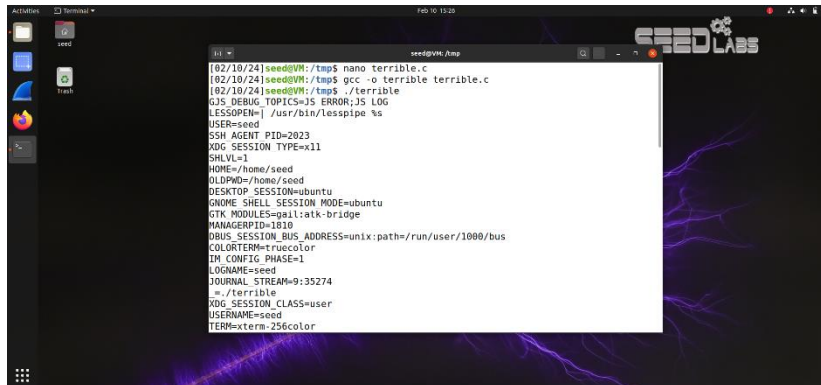
[02/05/24]seed@VM:~/../Labsetup$ myenv
[02/05/24]seed@VM:~/../Labsetup$ nano myenv.c
[02/05/24]seed@VM:~/../Labsetup$ gcc myenv.c -o myenv
[02/05/24]seed@VM:~/../Labsetup$ myenv
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1841,unix/VM:/tmp/.ICE-unix/1841
QT_ACCESSIBILITY=1
COLORTERM=truecolor
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=1794
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Downloads/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
GJS_DEBUG_TOPICS=JS ERROR;JS LOG

```

I compiled and ran the provided program, initially printing nothing. I then modified the program to include 'environ' in its 'execve' function arguments, printing out the environment variables.

4. Environment variables and system()

4.1 Screenshot and Summary



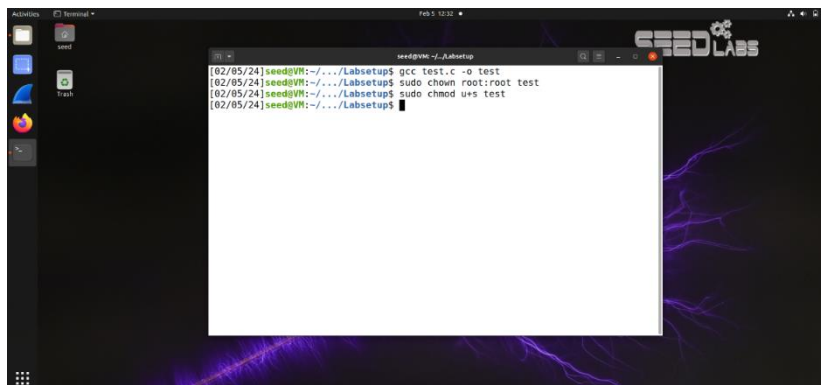
```

[02/10/24]seed@VR:~/tmp$ nano terrible.c
[02/10/24]seed@VR:~/tmp$ gcc -o terrible terrible.c
[02/10/24]seed@VR:~/tmp$ ./terrible
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
SSH_AGENT_PID=2023
XDG_SESSION_TYPE=x11
SHLVL=1
HOME=/home/seed
OLDPWD=/home/seed
DESKTOP_SESSION=ubuntu
XDG_RUNTIME_DIR=/run/user/1000
GTK_MODULES=gail:atk-bridge
MANAGERPID=1810
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
COLORTERM=truecolor
IM_CONFIG_PHASE=1
LOGNAME=seed
JOURNAL_STREAM=9:35274
././terrible
XDG_SESSION_CLASS=user
USERNAME=seed
TERM=xterm-256color
  
```

The system("/usr/bin/env") function in the provided program executes the /bin/sh -c command, asking the shell to execute the command. The system() function uses execl() to execute /bin/sh; execl() calls execve(), passing to it the environment variables array and printing them out.

5. Environment variable and Set-UID Programs

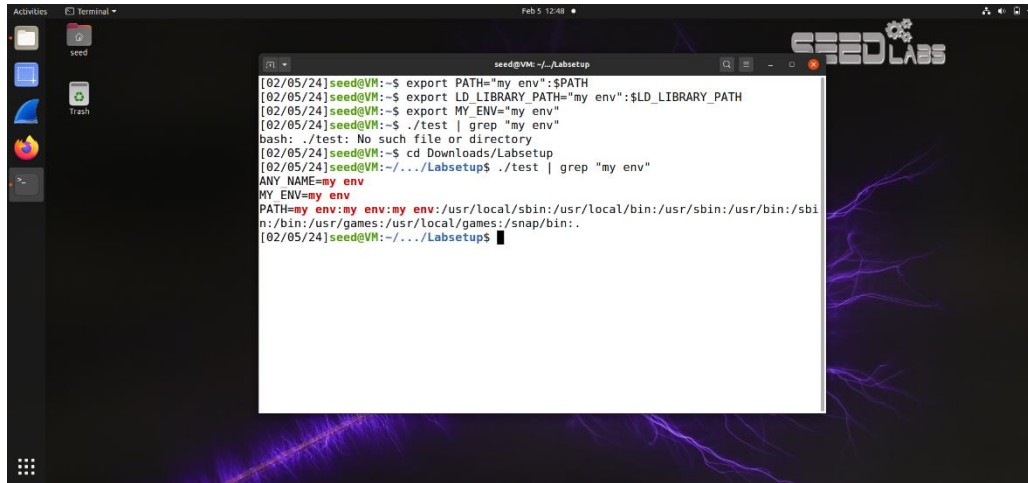
5.1 Screenshots and Summaries



```

[02/05/24]seed@VR:~/Labsetup$ gcc test.c -o test
[02/05/24]seed@VR:~/Labsetup$ sudo chown root:root test
[02/05/24]seed@VR:~/Labsetup$ sudo chmod u+s test
[02/05/24]seed@VR:~/Labsetup$
  
```

I simply compiled and ran the provided program and set the appropriate root ownership and Set-UID mode.



```

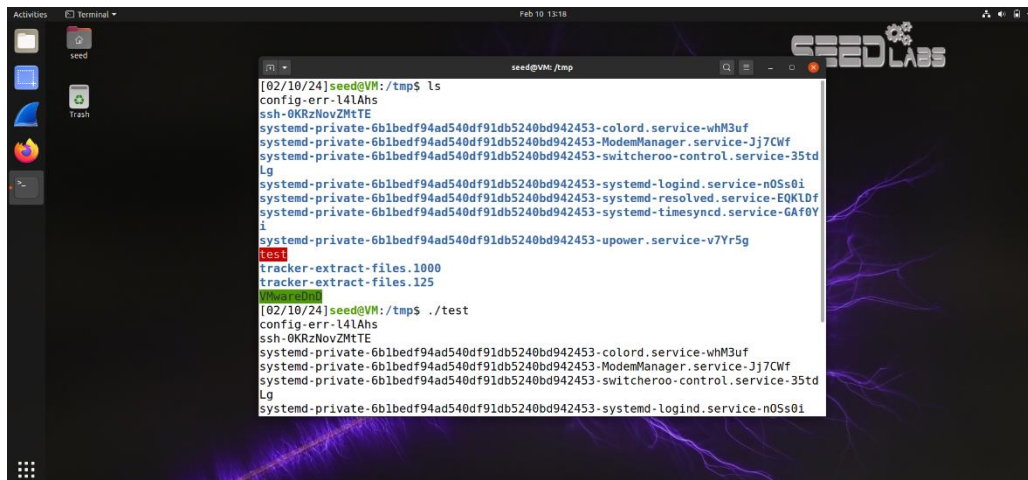
[02/05/24]seed@VM:~$ export PATH="my env":$PATH
[02/05/24]seed@VM:~$ export LD_LIBRARY_PATH="my env":$LD_LIBRARY_PATH
[02/05/24]seed@VM:~$ export MY_ENV="my env"
[02/05/24]seed@VM:~$ ./test | grep "my env"
bash: ./test: No such file or directory
[02/05/24]seed@VM:~$ cd Downloads/Labsetup
[02/05/24]seed@VM:~/Downloads/Labsetup$ ./test | grep "my env"
ANY_NAME=my env
MY_ENV=my env
PATH=my env:my env:my env:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:
[02/05/24]seed@VM:~/Downloads/Labsetup$

```

I proceeded to set the appropriate paths, run the program, and display the corresponding environment variables.

6. The PATH Environment variable and Set-UID Programs

6.1 Screenshot and Summary



```

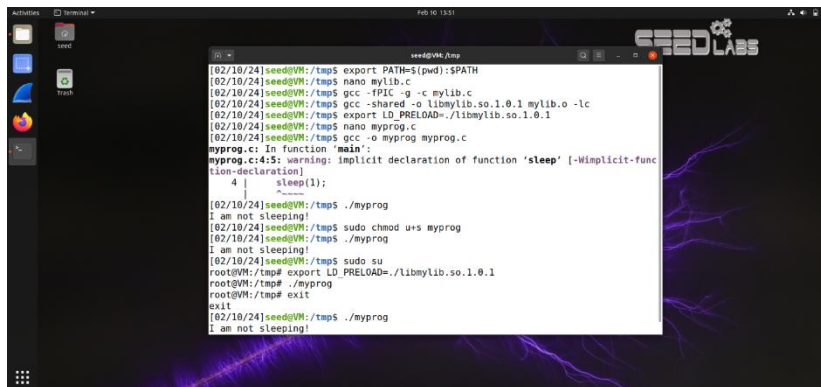
[02/10/24]seed@VM:/tmp$ ls
config-err-l4lAhs
ssh-0KRzNovZMtE
systemd-private-6b1bedf94ad540df91db5240bd942453-color.service-whM3uf
systemd-private-6b1bedf94ad540df91db5240bd942453-ModemManager.service-Jj7CWf
systemd-private-6b1bedf94ad540df91db5240bd942453-switcheroo-control.service-35tdLg
systemd-private-6b1bedf94ad540df91db5240bd942453-systemd-logind.service-n0Ss0i
systemd-private-6b1bedf94ad540df91db5240bd942453-systemd-resolved.service-EQKL0f
systemd-private-6b1bedf94ad540df91db5240bd942453-systemd-timesyncd.service-GA70Yi
systemd-private-6b1bedf94ad540df91db5240bd942453-upower.service-v7Yr5g
test
tracker-extract-files.1000
tracker-extract-files.125
VMusr00m
[02/10/24]seed@VM:/tmp$ ./test
config-err-l4lAhs
ssh-0KRzNovZMtE
systemd-private-6b1bedf94ad540df91db5240bd942453-color.service-whM3uf
systemd-private-6b1bedf94ad540df91db5240bd942453-ModemManager.service-Jj7CWf
systemd-private-6b1bedf94ad540df91db5240bd942453-switcheroo-control.service-35tdLg
systemd-private-6b1bedf94ad540df91db5240bd942453-systemd-logind.service-n0Ss0i

```

I compiled the provided program, changed the ownership to root, and made it Set-UID. I then displayed the contents of the current directory using the system(ls) function provided in the program.

7. The LD_PRELOAD environment variable and Set-UID Programs

7.1 Screenshots and Summaries

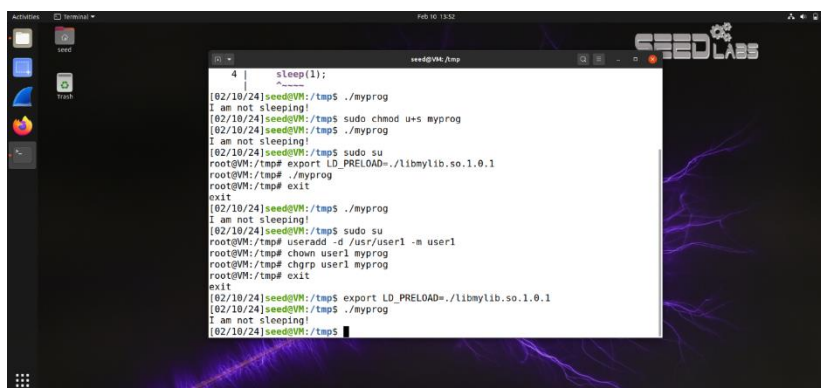


```

[02/10/24]seed@VM:/tmp$ export PATH=$PWD:$SPATH
[02/10/24]seed@VM:/tmp$ nano mylib.c
[02/10/24]seed@VM:/tmp$ gcc -fPIC -g -c mylib.c
[02/10/24]seed@VM:/tmp$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[02/10/24]seed@VM:/tmp$ export LD_PRELOAD=../libmylib.so.1.0.1
[02/10/24]seed@VM:/tmp$ nano myprog.c
[02/10/24]seed@VM:/tmp$ gcc -o myprog myprog.c
myprog.c: In function 'main':
myprog.c:4:5: warning: implicit declaration of function 'sleep' [-Wimplicit-func
tion-declaration]
    4 |     sleep(1);
      |     ^~~~~
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$ sudo chmod u+s myprog
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$ sudo su
root@VM:/tmp# export LD_PRELOAD=../libmylib.so.1.0.1
root@VM:/tmp# ./myprog
root@VM:/tmp# exit
exit
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!

```

I built a dynamic link library using the provided program mylib.c, compiling and setting the appropriate environment variables. I then compiled and ran another provided function myprog.c that executed the linked program in some cases and myprog.c in other cases. I believe that the output makes sense for the regular user and with Set-UID privileges as they are the one who set the environment variable that specifies mylib.c. It also makes sense for no output as root as to prevent unauthorized code from running.



```

    4 |     sleep(1);
      |     ^~~~~
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$ sudo chmod u+s myprog
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$ sudo su
root@VM:/tmp# export LD_PRELOAD=../libmylib.so.1.0.1
root@VM:/tmp# ./myprog
root@VM:/tmp# exit
exit
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$ sudo su
root@VM:/tmp# useradd -d /usr/user1 -m user1
root@VM:/tmp# chown user1 myprog
root@VM:/tmp# chgrp user1 myprog
root@VM:/tmp# exit
exit
[02/10/24]seed@VM:/tmp$ export LD_PRELOAD=../libmylib.so.1.0.1
[02/10/24]seed@VM:/tmp$ ./myprog
I am not sleeping!
[02/10/24]seed@VM:/tmp$

```

Finally, changing the program's ownership to another user, it still ran which was surprising. I suppose that the child processes inherit LD_* environment variables differently.

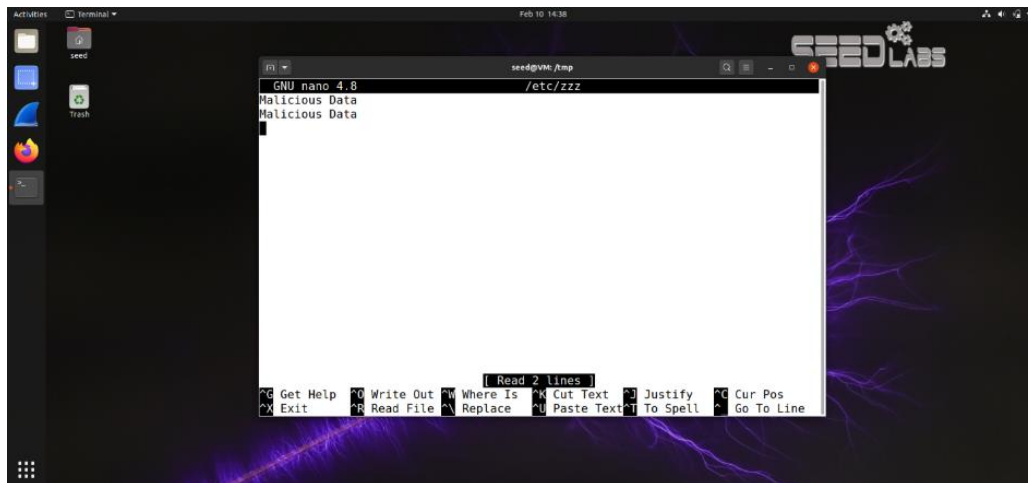
8. Invoking external program using system() versus execve()

8.1 Summary

When the program uses `system(command)` to invoke the command, it's vulnerable to various forms of attacks due to the way `system()` invokes a shell to execute the command. This shell can interpret shell metacharacters with root privileges due to the Set-UID bit., allowing an attacker like Bob to append additional commands to the input. The same attack wouldn't work if I switched the program to use `execve()` because `execve()` does not interpret the input as a shell command but rather as the path and arguments for a single executable.

9. Capability Leaking

9.1 Screenshot and Summary



In this final task, I compiled and ran the provided program as a normal user and was able to write to a root-owned file.