

Mini Project 3: Set-UID Program Vulnerability

Task 1: Explore SetUID Programs

Question 1. Did the programs work appropriately in both cases? Please briefly justify your observations.

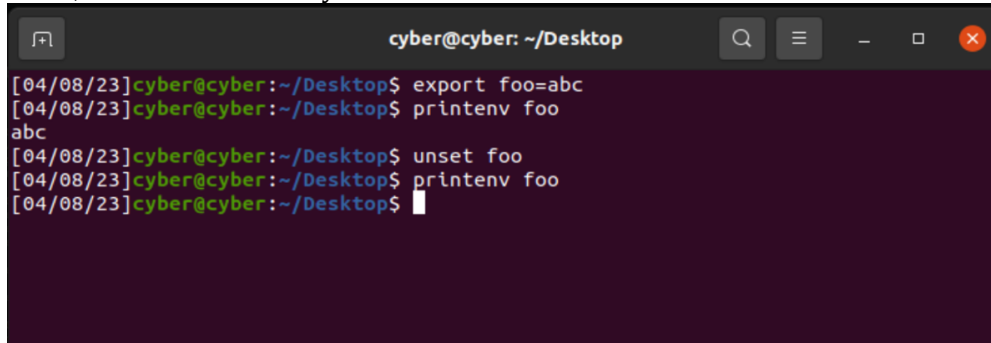
- Yes, the programs work appropriately in both cases. From my observations I can say that the Set-UID programs: **passwd**, **chsh**, and **sudo** work the same way no matter which directories they are in.

```
cyber@cyber: ~  
[04/08/23]cyber@cyber:~$ chsh  
Password:  
Changing the login shell for cyber  
Enter the new value, or press ENTER for the default  
Login Shell [/bin/bash]:  
[04/08/23]cyber@cyber:~$ sudo  
usage: sudo -h | -K | -k | -V  
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]  
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]  
[command]  
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]  
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-T timeout] [-u user] file ...  
[04/08/23]cyber@cyber:~$ passwd  
Changing password for cyber.  
Current password:  
New password:  
Retype new password:  
passwd: password updated successfully  
[04/08/23]cyber@cyber:~$
```

```
cyber@cyber: ~/Desktop  
[04/08/23]cyber@cyber:~$ cd Desktop/  
[04/08/23]cyber@cyber:~/Desktop$ sudo  
usage: sudo -h | -K | -k | -V  
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]  
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]  
[command]  
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]  
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-T timeout] [-u user] file ...  
[04/08/23]cyber@cyber:~/Desktop$ chsh  
Password:  
Changing the login shell for cyber  
Enter the new value, or press ENTER for the default  
Login Shell [/bin/bash]:  
[04/08/23]cyber@cyber:~/Desktop$ passwd  
Changing password for cyber.  
Current password:  
New password:  
Retype new password:  
passwd: password updated successfully  
[04/08/23]cyber@cyber:~/Desktop$
```

Task 2: Exploring Environment Variables


Question 2. Please set an environment variable called “foo” with a value of your choice, show its value, and unset it. Show your results with screenshots.



```
cyber@cyber: ~/Desktop
[04/08/23]cyber@cyber:~/Desktop$ export foo=abc
[04/08/23]cyber@cyber:~/Desktop$ printenv foo
abc
[04/08/23]cyber@cyber:~/Desktop$ unset foo
[04/08/23]cyber@cyber:~/Desktop$ printenv foo
[04/08/23]cyber@cyber:~/Desktop$
```

2.2 Passing Environment Variables from Parent Process to Child Process

Question 3. Compare the difference of the two output files using the diff command. Please describe your observations.



```
c206n885@cycle2:~/Desktop/MP3$ rm test1
c206n885@cycle2:~/Desktop/MP3$ rm test2
c206n885@cycle2:~/Desktop/MP3$ rm myprintenv
c206n885@cycle2:~/Desktop/MP3$ gcc myprintenv.c -o myprintenv
c206n885@cycle2:~/Desktop/MP3$ ./myprintenv > test1
c206n885@cycle2:~/Desktop/MP3$ gcc myprintenv.c -o myprintenv
c206n885@cycle2:~/Desktop/MP3$ ./myprintenv > test2
c206n885@cycle2:~/Desktop/MP3$ diff test1 test2
c206n885@cycle2:~/Desktop/MP3$
```

- After following the instruction, there was no difference in both outputs. So, we can say that the parent’s environment variables are inherited by the child process.

2.3 Environment Variables and `execve()`

Question 4. How does the new program get its environment variables? Please explain based on your observations.



```
#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
    execve("/usr/bin/env", argv, environ);
    return 0 ;
}
```

- Since the original **myenv.c** file have the **execve()** function with the **envp** argument set to **NULL** which mean that the child process create by **fork()** has no access to the environment

variables of the parent process. As a result, there is no output produced when the program is run.

- On the other hand, the second **execve()** function have the **environ** variable to pass the parent environment to the child process when called. This allows the child process to access the environment variables. Here the output

```
SHELL=/bin/bash
COLORTERM=truecolor
TERM_PROGRAM_VERSION=1.77.2
GTK_MODULES=appmenu-gtk-module:appmenu-gtk-module
PWD=/home/c206n885/Desktop/MP3
KRB5CCNAME=FILE:/tmp/krb5cc_100599096_JcJy9b
LOGNAME=c206n885
XDG_SESSION_TYPE=ttty
VSCODE_GIT_ASKPASS_NODE=/home/c206n885/.vscode-
server/bin/e344f1f539a80912a0e9357cec841f36ce97a4e2/node
MOTD_SHOWN=pam
HOME=/home/c206n885
LANG=en_US.UTF-8
GIT_ASKPASS=/home/c206n885/.vscode-
server/bin/e344f1f539a80912a0e9357cec841f36ce97a4e2/extensions/git/dist/askpass.sh
SSH_CONNECTION=194.156.136.4 51378 129.237.87.112 22
foo=abc
VSCODE_GIT_ASKPASS_EXTRA_ARGS=
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=c206n885
VSCODE_GIT_IPC_HANDLE=/run/user/100599096/vscode-git-de526c64c0.sock
SHLVL=2
UBUNTU_MENUPROXY=1
XDG_SESSION_ID=148328
XDG_RUNTIME_DIR=/run/user/100599096
SSH_CLIENT=194.156.136.4 51378 22
VSCODE_GIT_ASKPASS_MAIN=/home/c206n885/.vscode-
server/bin/e344f1f539a80912a0e9357cec841f36ce97a4e2/extensions/git/dist/askpass-
main.js
XDG_DATA_DIRS=/usr/share/gnome:/home/c206n885/.local/share/flatpak/exports/share:/var/
lib/flatpak/exports/share:/usr/local/share:/usr/share:/var/lib/snapd/desktop
BROWSER=/home/c206n885/.vscode-
server/bin/e344f1f539a80912a0e9357cec841f36ce97a4e2/bin/helpers/browser.sh
PATH=/home/c206n885/.vscode-
server/bin/e344f1f539a80912a0e9357cec841f36ce97a4e2/bin/remote-cli:/opt/node-v16.17.1-
linux-x64/bin:/opt/node-v16.17.1-linux-
x64/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin:/home/c206n885/bin:/home/c206n885/bin
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/100599096/bus
OLDPWD=/home/c206n885/Desktop
TERM_PROGRAM=vscode
```

```
VSCODE_IPC_HOOK_CLI=/run/user/100599096/vscode-ipc-ad6651a4-8018-4077-9f73-77db2ed06102.sock
_=. /myenv
```

2.4 Environment Variables and system()

Question 5. How does the new program `/bin/sh` get its environment variables? Please explain based on your observations.

```
c206n885@cycle2:~/Desktop/MP3$ gcc mysystem.c -o mys
c206n885@cycle2:~/Desktop/MP3$ ./mys > case1
c206n885@cycle2:~/Desktop/MP3$ export foo=abc
c206n885@cycle2:~/Desktop/MP3$ gcc mysystem.c -o mys
c206n885@cycle2:~/Desktop/MP3$ ./mys > case2
c206n885@cycle2:~/Desktop/MP3$ diff case1 case2
8a9
> foo=abc
c206n885@cycle2:~/Desktop/MP3$
```

- The new program **/bin/sh** gets its environment variables from the parent process that called **system()**. When **system()** is called, it executes **/bin/sh** and ask the shell to execute the command. The shell process, in turn inherits the environment variables of the parent process, which is the program that called system. Since the shell process can modify or add new environment variables which will be inherited by any subsequent processes it executes. So, after **export foo=abc** being executes, it will be available and be inherited by the shell command.

Task 3: Environment Variables and Set-UID Programs

When a Set-UID program runs, it assumes the owner's privileges. Therefore, Set-UID programs could result in privilege escalation. It is quite risky despite being useful in many tasks.

3.1 Use Environment Variables to Affect Set-UID Programs

A screenshot of a terminal window titled "cyber@cyber: ~/Desktop". The terminal shows the command "cat /etc/environment" being executed, resulting in a long list of environment variables and their values. The variables include system paths like "/usr/bin" and "/usr/lib", configuration directories like "/etc/xdg", session identifiers like "GNOME_DESKTOP_SESSION_ID", and user-specific settings like "HOME" and "USERNAME". The terminal window has a dark background and a sidebar on the left with icons for "Activities", "Terminal", and "Files". The top of the window shows the system clock as "Apr 18 2025".

- By using the command **sudo ln -sf /bin/zsh/bin/sh** we can view the content of the **shadow** file.

```
cyber@cyber: ~/.../MP3
[04/13/23] cyber@cyber: ~/.../MP3$ sudo ln -sf /bin/zsh /bin/sh
[04/13/23] cyber@cyber: ~/.../MP3$ ./nyle
root::19081:0:9999:7:::
daemon:*:19046:0:9999:7:::
bin:*:19046:0:9999:7:::
sys:*:19046:0:9999:7:::
sync:*:19046:0:9999:7:::
games:*:19046:0:9999:7:::
man:*:19046:0:9999:7:::
lp:*:19046:0:9999:7:::
mail:*:19046:0:9999:7:::
news:*:19046:0:9999:7:::
uucp:*:19046:0:9999:7:::
proxy:*:19046:0:9999:7:::
www-data:*:19046:0:9999:7:::
backup:*:19046:0:9999:7:::
l1st:*:19046:0:9999:7:::
lrc:*:19046:0:9999:7:::
gnats:*:19046:0:9999:7:::
nobody:*:19046:0:9999:7:::
systemd-network:*:19046:0:9999:7:::
systemd-resolve:*:19046:0:9999:7:::
systemd-timesync:*:19046:0:9999:7:::
messagebus:*:19046:0:9999:7:::
syslog:*:19046:0:9999:7:::
_apt:*:19046:0:9999:7:::
ts:*:19046:0:9999:7:::
uuldd:*:19046:0:9999:7:::
tcpdump:*:19046:0:9999:7:::
avahi-autotpd:*:19046:0:9999:7:::
usbmux:*:19046:0:9999:7:::
rtkit:*:19046:0:9999:7:::
dnsmasq:*:19046:0:9999:7:::
cups-pk-helper:*:19046:0:9999:7:::
speech-dispatcher:*:19046:0:9999:7:::
avahi:*:19046:0:9999:7:::
kernoops:*:19046:0:9999:7:::
saned:*:19046:0:9999:7:::
nm-openvpn:*:19046:0:9999:7:::
hplip:*:19046:0:9999:7:::
whoopie:*:19046:0:9999:7:::
colord:*:19046:0:9999:7:::
geoclue:*:19046:0:9999:7:::
pulse:*:19046:0:9999:7:::
gnome-initial-setup:*:19046:0:9999:7:::
gdm:*:19046:0:9999:7:::
sshd:*:19046:0:9999:7:::
cyber:$658nm20v2461syZSL/X2qn60tCqc/TC4u037Y9Cx7BEnShZ80F0Amk3qeEH/9oSun6cdsV8KrkSwgghvkDQ2_8sKTF76y9v1LCR0:19393:0:9999:7:::
systemd-coredump:!!!19081!!!!:
sshd:*:19234:0:9999:7:::
fwupd-refresh:*:19309:0:9999:7:::
telnetd:*:19391:0:9999:7:::
ftp:*:19391:0:9999:7:::
research:$65c90pY0bvPOBHaL..$.aT3nE6dqq52TYqfz..iFN63tJl12TMCAABpc3WRZn1f00KWPZ3J3ay7q3VNKkwtPFEZx4NQC4QfNF50/14LKAL0:19422:0:9999:7:::
[04/13/23] cyber@cyber: ~/.../MP3$
```