Lab1: SEED Labs – Environment Variable and Set-UID Program

Mayur Suresh

CSC 302-01: Computer Security

Date: February 12th, 2023

Task 1: Manipulating Environment Variables

```
seed@pcvm815-1:~/setuid_lab$ ls
Labsetup.zip
seed@pcvm815-1:~/setuid_lab$ unzip Labsetup.zip
Archive:  Labsetup.zip
   creating: Labsetup/
  inflating: Labsetup/.cap_leak.c.swp
  inflating: Labsetup/cap_leak.c
  inflating: Labsetup/myenv.c
  inflating: Labsetup/myprintenv.c
  inflating: Labsetup/catall.c
seed@pcvm815-1:~/setuid_lab$
```

The above picture shows the files under the Labsetup directory downloaded from the link 'https://seedsecuritylabs.org/Labs_20.04/Files/Environment_Variable_and_SetUID/Labsetup.zip'

Using the seed account we use the env and the printenc command to print out the environment variables. Below we see both the env and the printenv commands used both giving us the environment variables.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ env
SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=104.201.188.52 51004 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
XDG_SESSION_ID=803
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=104.201.188.52 51004 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
n:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
_=/usr/bin/env
OLDPWD=/home/seed/setuid_lab
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ printenv
SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=104.201.188.52 51004 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
XDG_SESSION_ID=803
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=104.201.188.52 51004 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
n:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
_=/usr/bin/printenv
OLDPWD=/home/seed/setuid_lab
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

We can see specific environment variables by specifically inputting the variable as shown bellow with SHELL and PWD.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ printenv SHELL
/bin/bash
seed@pcvm815-1:~/setuid_lab/Labsetup$ env | grep PWD
PWD=/home/seed/setuid_lab/Labsetup
OLDPWD=/home/seed/setuid_lab
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

Next we look at the file myenv.c with the `cat` command to check the file, then we check the PATH with the `echo $PATH` command. We see that there is still the path, which is still not reset and still showing the path address we used in the last lab.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ ls
cap_leak.c  catall.c  myenv  myenv.c  myprintenv.c
seed@pcvm815-1:~/setuid_lab/Labsetup$ cat myenv.c
#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);

  return 0 ;
}
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr
/local/games:.
```

The next steps are to set up the countermeasures for the EV's dynamic linking as discussed in class, which includes the steps below.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ export LD_PRELOAD=./libmylib.so.1.0.1
seed@pcvm815-1:~/setuid_lab/Labsetup$ export LD_LIBRARY_PATH=.
seed@pcvm815-1:~/setuid_lab/Labsetup$ export LD_MYOWN="my own EV"
seed@pcvm815-1:~/setuid_lab/Labsetup$ env
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
LD_PRELOAD=./libmylib.so.1.0.1
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=104.201.188.52 51004 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
LD_MYOWN=my own EV
XDG_SESSION_ID=803
LD_LIBRARY_PATH=.
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=104.201.188.52 51004 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
OLDPWD=/home/seed/setuid_lab
_=/usr/bin/env
seed@pcvm815-1:~/setuid_lab/Labsetup$ env | grep LD_
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
LD_PRELOAD=./libmylib.so.1.0.1
LD_MYOWN=my own EV
LD_LIBRARY_PATH=.
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ myenv | grep LD_
ERROR: ld.so: object './libmylib.so.1.0.1ERROR: ld.so: object './libmylib.s
o.1.0.1' from LD_PRELOAD cannot be preloaded (cannot open shared object fil
e): ignored.
' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ign
ored.
LD_PRELOAD=./libmylib.so.1.0.1
LD_MYOWN=my own EV
LD_LIBRARY_PATH=.
```

I run into the error shown below and this is because the LD_PRELOAD still has the path from a previous exercise.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ sudo chown root myenv
[sudo] password for seed:
seed@pcvm815-1:~/setuid_lab/Labsetup$ sudo chmod root myenv
chmod: invalid mode: ‘root’
Try 'chmod --help' for more information.
seed@pcvm815-1:~/setuid_lab/Labsetup$ sudo chmod 4755 myenv
seed@pcvm815-1:~/setuid_lab/Labsetup$ myenv | grep LD_
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloa
ded (cannot open shared object file): ignored.
LD_MYOWN=my own EV
seed@pcvm815-1:~/setuid_lab/Labsetup$ ls
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_PRELOAD cannot be preloa
ded (cannot open shared object file): ignored.
cap_leak.c  catall.c  myenv  myenv.c  myprintenv.c
seed@pcvm815-1:~/setuid_lab/Labsetup$ export  PATH_PRELOAD =""
-bash: export: `=': not a valid identifier
seed@pcvm815-1:~/setuid_lab/Labsetup$ export  LD_PRELOAD=""
seed@pcvm815-1:~/setuid_lab/Labsetup$ myenv | grep LD_
LD_MYOWN=my own EV
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

Because LD_PRELOAD was still pointing to the previous address we use the `export LD_PRELOAD` command to set it to an empty string.

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

For this task we will study how a child process gets its environment variables from its parent. We first compile the myprintenv.c file found under the Labsetup directory previously downloaded. With this we get binary file a.out and we see below what is under the file after running it we see the output below.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ cat myprintenv.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

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

void main()
{
  pid_t childPid;
  switch(childPid = fork()) {
    case 0:  /* child process */
      printenv();
      exit(0);
    default:  /* parent process */
      //printenv();
      exit(0);
  }
}
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ ./a.out
seed@pcvm815-1:~/setuid_lab/Labsetup$ SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=71.230.28.39 51002 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
XDG_SESSION_ID=1798
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=71.230.28.39 51002 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
_=./a.out
OLDPWD=/home/seed/setuid_lab
```

Next, we edit the myprintenv.c file and comment out the printenv() function under the child process and uncomment the function under the parent process. We then compile it again and we see the binary file reproduced and shown below.

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ cat myprintenv.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

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

void main()
{
  pid_t childPid;
  switch(childPid = fork()) {
    case 0:  /* child process */
      //printenv();
      exit(0);
    default:  /* parent process */
      printenv();
      exit(0);
  }
}
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ ./a.out
SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=71.230.28.39 51002 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
XDG_SESSION_ID=1798
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=71.230.28.39 51002 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
_=./a.out
OLDPWD=/home/seed/setuid_lab
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

Finally, we compare the two environment variables and we see that there is no difference between them. This is because of the fork() that creates a child process which is essentially a copy of the parent process which inherits all the environment variables of the parent process. This makes them essentially the same thing at this creation. The newly created child process has a default of 0 which the why the switch statement works even without the printenv function that was commented out.

Task 3: Environment Variables and execve()

For this task we will see how the environment variables are affected when a new program is executed with the execve(). This function calls a system call to load new commands executes it.

First we compile and run the myenv.c program with the command `gcc myenv.c`. This program executes a program called `/usr/bin/env/`.

```
#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);

  return 0 ;
}
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ ./myenv.c
-bash: ./myenv.c: Permission denied
```

After running and compiling the c file we get a `Permission denied' message. This might be due to the NULL parameter for the execve function. We move forward to changing the NULL parameter under the execve function to `environ` from the **environ variable. We go head and run it.

```
#include <unistd.h>

extern char **environ;

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

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

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

  return 0 ;
}

seed@pcvm815-1:~/setuid_lab/Labsetup$
```

```
seed@pcvm815-1:~/setuid_lab/Labsetup$ ./myenv
SHELL=/bin/bash
PWD=/home/seed/setuid_lab/Labsetup
LOGNAME=seed
XDG_SESSION_TYPE=tty
MOTD_SHOWN=pam
HOME=/home/seed
LANG=en_US.UTF-8
SSH_CONNECTION=71.230.28.39 51002 155.98.37.71 22
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=seed
SHLVL=1
XDG_SESSION_ID=1798
XDG_RUNTIME_DIR=/run/user/38503
SSH_CLIENT=71.230.28.39 51002 22
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/38503/bus
SSH_TTY=/dev/pts/0
_=./myenv
OLDPWD=/home/seed/setuid_lab
seed@pcvm815-1:~/setuid_lab/Labsetup$
```

Finally, we see the environment variables with the execve() function. After exchanging the NULL with the `environ` variable we are now able to access the environment variables.