

CS6570 Assignment 1b - Report

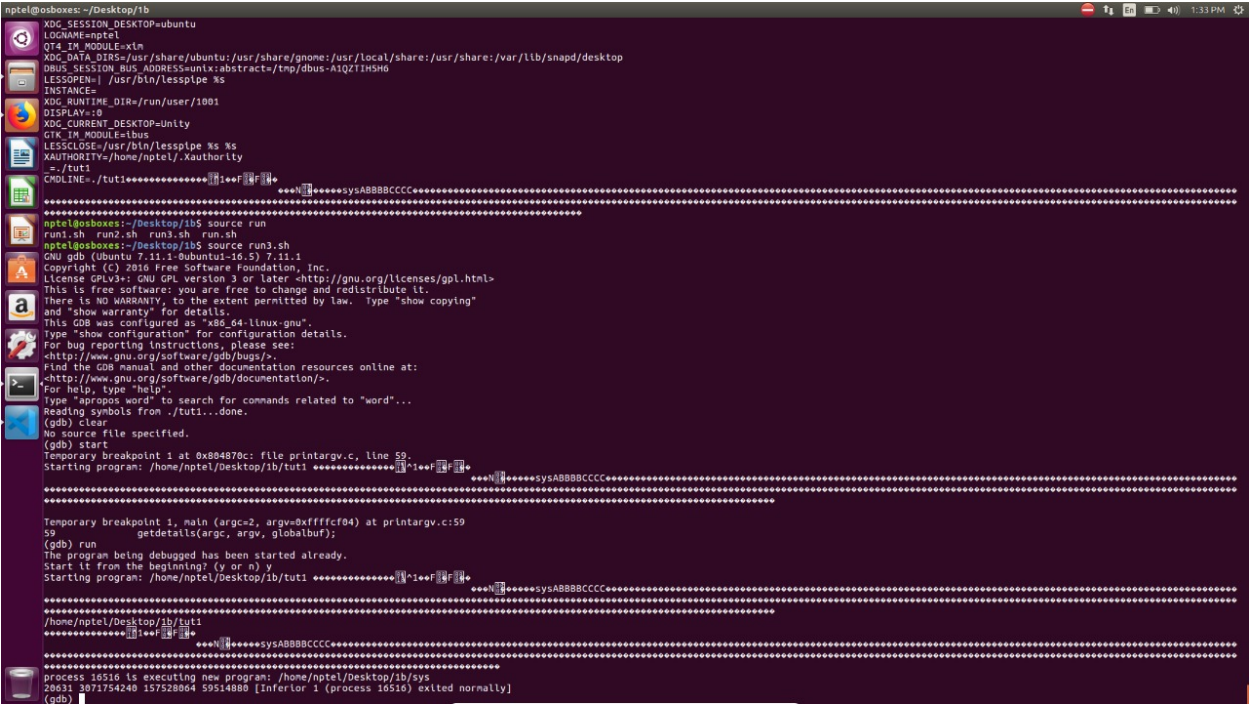
Rachit Tibrewal (CS16B022)
E Santhosh Kumar (CS16B107)

The printargv.c function is intended to take a sequence of command line arguments as input, and then construct a string with these arguments as well as the process’s environment variables.

We attack the program by subverting execution while returning from the "mergectmdline" function. This function copies the command line arguments into a local buffer of size 512 bytes. The function does not check if the size of arguments is greater than the allotted 512 bytes. Thus we use this vulnerability to over-write the return address and also inject our payload into the stack. Since we make the stack executable while compiling code, we can make the return address point back to the stack and execute the payload.

We assume that compiled executable of sysinfo.c is available in the same directory as that of the printargv executable. The payload uses the sys_execve system call (call number 11) to directly run the sysinfo executable. We call the system call in assembly code by putting the call number in the %eax register (and the corresponding executable name as argument in %ebx,).

The following screenshots show the subverted control flow and the final output.



(1).png

```
npotel@osboxes: ~/Desktop/1b
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
COMPIZ_CONFIG_PROFILE=ubuntu
COMMISSION=ubuntu
SESSIONTYPE=gnome-session
GTK_MODULES=overlay-scrollbar
XDG_SEAT=seat0
HOME=/home/npotel
SHLVL=1
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=npotel
QT4_IM_MODULE=xim
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/ gnome:/usr/local/share:/usr/share:/var/lib/ snapd/desktop
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-A1QZTINSH6
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
XDG_RUNTIME_DIR=/run/user/1001
DISPLAY=:0
XDG_CURRENT_DESKTOP=unity
GTK_IM_MODULE=ibus
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/npotel/.xauthority
~/tut1
CMDLINE=/tut1
***N***sysABBBBCCCC*****
npotel@osboxes:~/Desktop/1b$ source run
run1.sh run2.sh run3.sh run.sh
npotel@osboxes:~/Desktop/1b$ source run3.sh
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./tut1...done.
(gdb) clear
No source file specified.
(gdb) start
Temporary breakpoint 1 at 0x804870c: file printargv.c, line 59.
Starting program: /home/npotel/Desktop/1b/tut1 *****N*****
*****N*****
Temporary breakpoint 1, main (argc=2, argv=0xffffcf04) at printargv.c:59
59 getdetails(argc, argv, globalbuff);
(gdb)
```