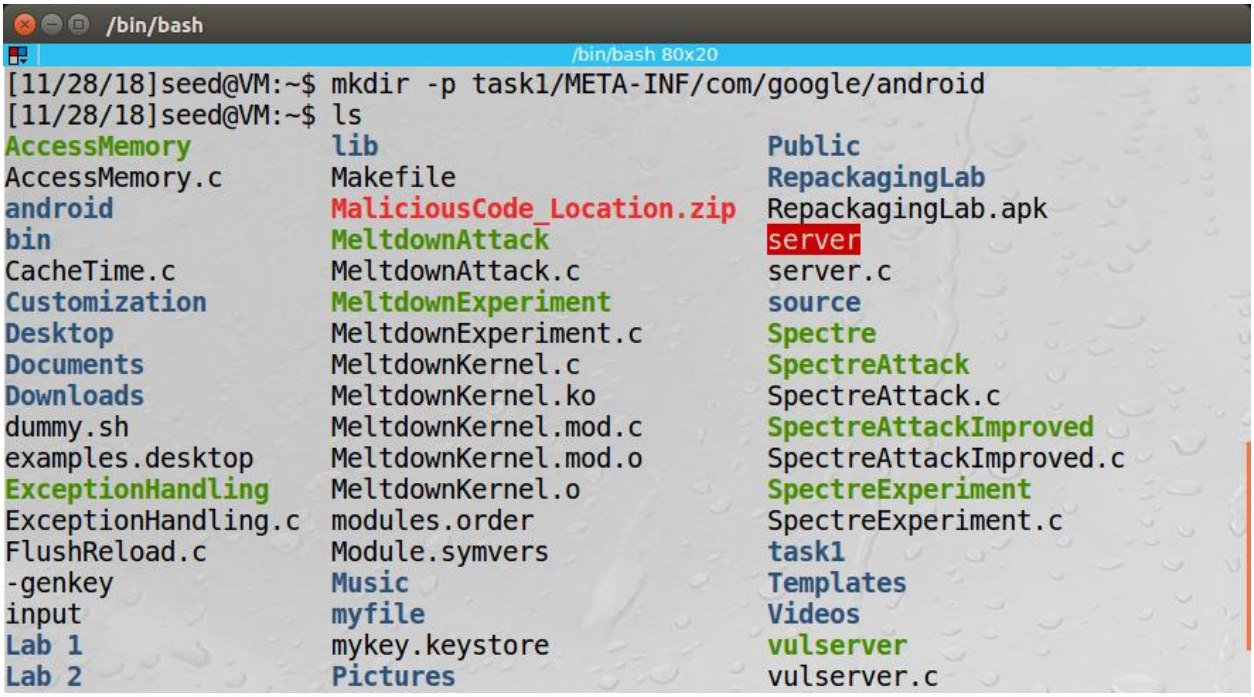
Android Rooting Device Lab

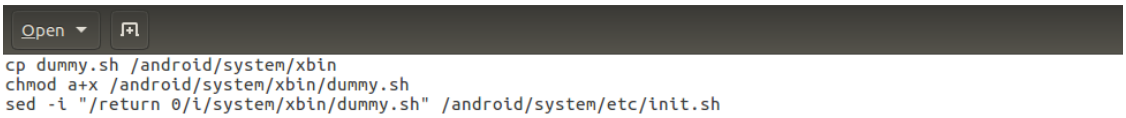
Name: Raman Srivastava
SUID: 946665605

TASK 1



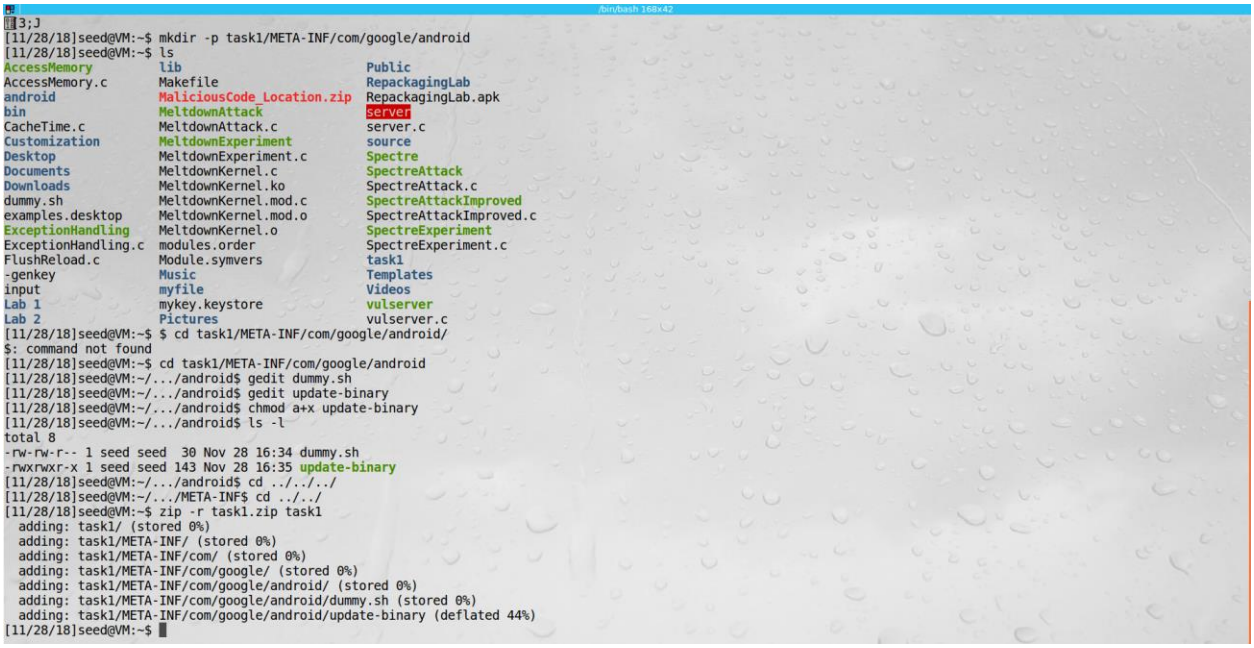
Here I did the mkdir -p task1/META-INF/com/google/android command to create OTP package

This is the script present in dummy.sh file



This is the update binary file that will copy the dummy.sh file into the xbin folder and replace the script before the return 0 of the init.sh script.

Here, we perform the chmod a+x update-binary



The ZIP file for the OTA has been created and now we'll send this to the android VM using the

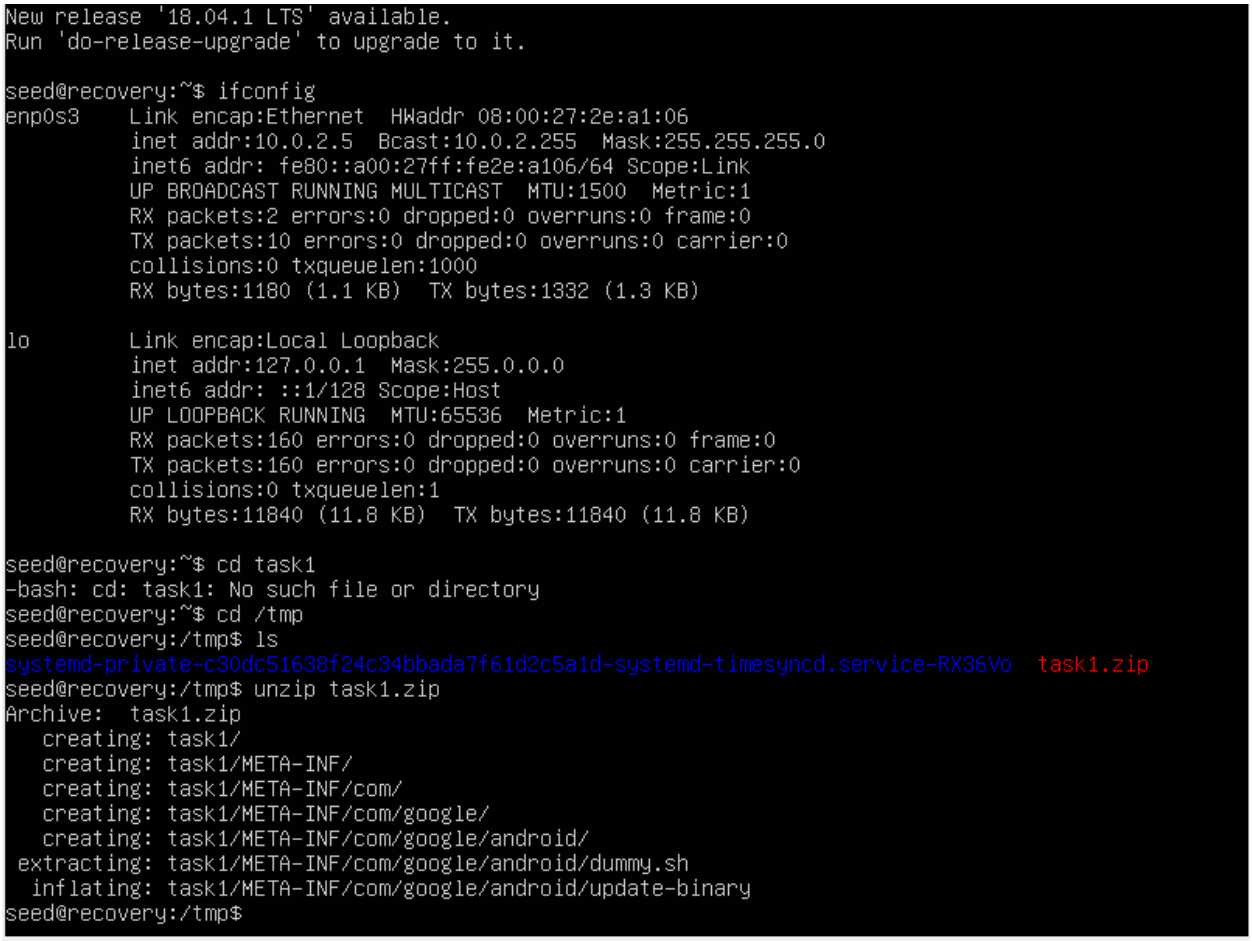
scp task1.zip seed@10.0.2.5:/tmp command

```
Ubuntu 16.04.4 LTS recovery tty1

recovery login: seed
Password:
Last login: Wed Nov 28 15:56:31 EST 2018 on tty1
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

seed@recovery:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:2e:a1:06
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2e:a106/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KB)  TX bytes:1332 (1.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

seed@recovery:~$ _
```

IP address after booting into recovery OS to send the OTA package using the scp command in the seed virtual machine

```
/bin/bash 168×42
Customization       MeltdownExperiment          source
Desktop             MeltdownExperiment.c        Spectre
Documents           MeltdownKernel.c            SpectreAttack
Downloads           MeltdownKernel.ko           SpectreAttack.c
dummy.sh            MeltdownKernel.mod.c        SpectreAttackImproved
examples.desktop    MeltdownKernel.mod.o        SpectreAttackImproved.c
ExceptionHandling   MeltdownKernel.o           SpectreExperiment
ExceptionHandling.c modules.order               SpectreExperiment.c
FlushReload.c       Module.symvers              task1
-genkey             Music                       Templates
input               myfile                      Videos
Lab 1               mykey.keystore              vulserver
Lab 2               Pictures                    vulserver.c
[11/28/18]seed@VM:~$ $ cd task1/META-INF/com/google/android/
$: command not found
[11/28/18]seed@VM:~$ cd task1/META-INF/com/google/android
[11/28/18]seed@VM:~/.../android$ gedit dummy.sh
[11/28/18]seed@VM:~/.../android$ gedit update-binary
[11/28/18]seed@VM:~/.../android$ chmod a+x update-binary
[11/28/18]seed@VM:~/.../android$ ls -l
total 8
-rw-rw-r-- 1 seed seed  30 Nov 28 16:34 dummy.sh
-rwxrwxr-x 1 seed seed 143 Nov 28 16:35 update-binary
[11/28/18]seed@VM:~/.../android$ cd ../../../
[11/28/18]seed@VM:~/.../META-INF$ cd ../../
[11/28/18]seed@VM:~$ zip -r task1.zip task1
  adding: task1/ (stored 0%)
  adding: task1/META-INF/ (stored 0%)
  adding: task1/META-INF/com/ (stored 0%)
  adding: task1/META-INF/com/google/ (stored 0%)
  adding: task1/META-INF/com/google/android/ (stored 0%)
  adding: task1/META-INF/com/google/android/dummy.sh (stored 0%)
  adding: task1/META-INF/com/google/android/update-binary (deflated 44%)
[11/28/18]seed@VM:~$ scp task1.zip seed@10.0.2.5:/tmp
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ECDSA key fingerprint is SHA256:j27XN+nmbyA0avocrLHpQPiGRIzknAWmJli5yO6vrsA.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '10.0.2.5' (ECDSA) to the list of known hosts.
seed@10.0.2.5's password:
task1.zip                                     100% 1406     1.4KB/s   00:00
[11/28/18]seed@VM:~$
```

We've sent the OTA task1.zip to the recovery OS using the scp command.



Here, we've gone to the /tmp folder of the recovery OS where we sent the OTA file and we've unzipped our task1.zip file.

```
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KB)  TX bytes:1332 (1.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

seed@recovery:~$ cd task1
-bash: cd: task1: No such file or directory
seed@recovery:~$ cd /tmp
seed@recovery:/tmp$ ls
systemd-private-c30dc51638f24c34bbada7f61d2c5a1d-systemd-timesyncd.service-RX36Vo  task1.zip
seed@recovery:/tmp$ unzip task1.zip
Archive:  task1.zip
   creating: task1/
   creating: task1/META-INF/
   creating: task1/META-INF/com/
   creating: task1/META-INF/com/google/
   creating: task1/META-INF/com/google/android/
 extracting: task1/META-INF/com/google/android/dummy.sh
  inflating: task1/META-INF/com/google/android/update-binary
seed@recovery:/tmp$  cd /tmp/task1/META-INF/com/google/android
seed@recovery:/tmp/task1/META-INF/com/google/android$ ls
dummy.sh  update-binary
seed@recovery:/tmp/task1/META-INF/com/google/android$ ./update-binary
cp: cannot create regular file '/android/system/xbin/dummy.sh': Permission denied
chmod: cannot access '/android/system/xbin/dummy.sh': No such file or directory
sed: couldn't open temporary file /android/system/etc/sedQJ9kCB: Permission denied
seed@recovery:/tmp/task1/META-INF/com/google/android$ sudo ./update-binary
[sudo] password for seed:
seed@recovery:/tmp/task1/META-INF/com/google/android$ sudo reboot
```

Now we run the update-binary

We can see that the update-binary script has been executed and dummy.sh file is present in the ~/system/xbin directory.

TASK 2

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
extern char** environ;
int main(int argc, char** argv) {
//Write the dummy file
FILE* f = fopen("/system/dummy2", "w");
if (f == NULL) {
printf("Permission Denied.\n");
exit(EXIT_FAILURE);
}
fclose(f);
//Launch the original binary
char* cmd = "/system/bin/app_process_original";
execve(cmd, argv, environ);
//execve() returns only if it fails
return EXIT_FAILURE;
}
```

This is the app_process.c program that we'll compile using NDK.

```
APP_ABI := x86
APP_PLATFORM := android-21
APP_STL := stlport_static
APP_BUILD_SCRIPT := Android.mk
```

This is the Application.mk file

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := app_process64
LOCAL_SRC_FILES := app_process.c
include $(BUILD_EXECUTABLE)
```

This is the Android.mk where we've filled in the LOCAL_MODULE and LOCAL_SRC_FILES field.

```
/bin/bash
/bin/bash 80x24
[11/30/18]seed@VM:~$ cd task2_code
[11/30/18]seed@VM:~/task2_code$ ls
Android.mk  Application.mk  my_app_process.c
[11/30/18]seed@VM:~/task2_code$ gedit Application.mk
[11/30/18]seed@VM:~/task2_code$ gedit Android.mk
[11/30/18]seed@VM:~/task2_code$ gcc -o my_app_process my_app_process.c
[11/30/18]seed@VM:~/task2_code$ ls
Android.mk  Application.mk  my_app_process  my_app_process.c
[11/30/18]seed@VM:~/task2_code$ ./my_app_process
Permission Denied.
[11/30/18]seed@VM:~/task2_code$ export NDK_PROJECT_PATH=.
[11/30/18]seed@VM:~/task2_code$ ndk-build NDK_APPLICATION_MK=./Application.mk
Compile x86    : my_app_process <= my_app_process.c
Executable     : my_app_process
Install        : my_app_process => libs/x86/my_app_process
[11/30/18]seed@VM:~/task2_code$ ls
Android.mk  Application.mk  libs  my_app_process  my_app_process.c  obj
[11/30/18]seed@VM:~/task2_code$ cd libs
[11/30/18]seed@VM:~/.../libs$ ls
x86
[11/30/18]seed@VM:~/.../libs$ cd x86
[11/30/18]seed@VM:~/.../x86$ ls
my_app_process
[11/30/18]seed@VM:~/.../x86$
```

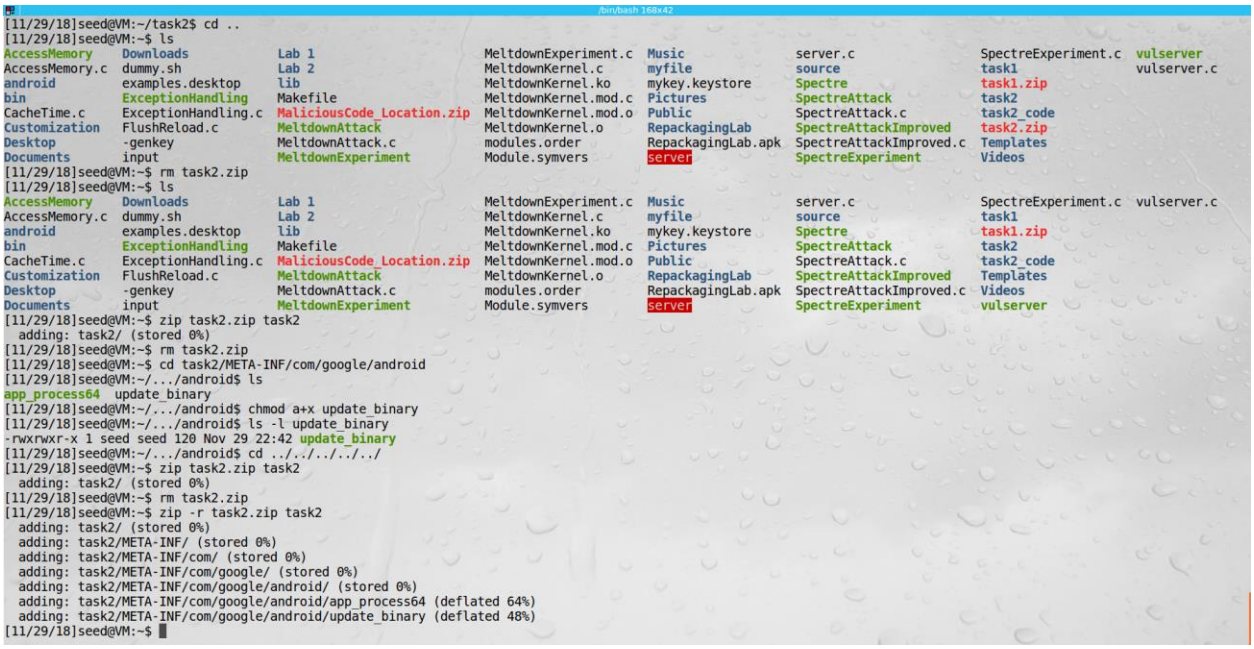
We got the binary file in libs/x86/my_app_process

```
[11/30/18]seed@VM:~$ cd task2_code
[11/30/18]seed@VM:~/task2_code$ ls
Android.mk  Application.mk  libs  my_app_process  my_app_process.c  obj
[11/30/18]seed@VM:~/task2_code$ cd ..
[11/30/18]seed@VM:~$ mv ~/task2_code/my_app_process ~/task2/META-INF/com/google/
android
[11/30/18]seed@VM:~$ cd task2/META-INF/com/google/android/
[11/30/18]seed@VM:~/.../android$ ls
my_app_process
[11/30/18]seed@VM:~/.../android$
```

We created a directory for task2/META-INF/com/google/android and we've moved app_process to it.

```
Open
mv /android/system/bin/app_process64 /android/system/bin/app_process_original
cp app_process64 /android/system/bin/app_process64
chmod a+x /android/system/bin/app_process64
```

This is the update-binary file where we rename app_process64 file to app_process_original that will be called later in the c program.



We've compressed task2 folder that we'll send to the recovery OS over SCP

```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

seed@recovery:~$ ls
seed@recovery:~$ cd ..
seed@recovery:/home$ ls
seed
seed@recovery:/home$ cd ..
seed@recovery:/$ ls
android  boot  etc   initrd.img  lib64       media  opt   root  sbin  sys  usr  vmlinuz
bin      dev   home  lib         lost+found  mnt    proc  run   srv   tmp  var
seed@recovery:/$ cd tmp
seed@recovery:/tmp$ ls
systemd-private-6dcd0d82de4447ed8e465b109803768b-systemd-timesyncd.service-1m6dcj  task2.zip
seed@recovery:/tmp$ unzip task2.zip
Archive:  task2.zip
   creating: task2/
   creating: task2/META-INF/
   creating: task2/META-INF/com/
   creating: task2/META-INF/com/google/
   creating: task2/META-INF/com/google/android/
  inflating: task2/META-INF/com/google/android/app_process64
  inflating: task2/META-INF/com/google/android/update_binary
seed@recovery:/tmp$ sudo ./update_binary
[sudo] password for seed:
sudo: ./update_binary: command not found
seed@recovery:/tmp$ cd /task2/META-INF/com/google/android
-bash: cd: /task2/META-INF/com/google/android: No such file or directory
seed@recovery:/tmp$ cd task2/META-INF/com/google/android
seed@recovery:/tmp/task2/META-INF/com/google/android$ ls
app_process64  update_binary
seed@recovery:/tmp/task2/META-INF/com/google/android$ sudo ./update_binary
seed@recovery:/tmp/task2/META-INF/com/google/android$
```

Here we can see that dummy2 file has been created in /android/system folder making this task successful.

TASK 3

The app_process64 file that's in this update-binary is a server program that waits for a client to connect to it. Once a client is connected, it spawns a child process and deligates the I/O file descriptors of the client to the child and launches the root shell.

```
seed@recovery:~$ cd /tmp/
seed@recovery:/tmp$ ls -l
total 20
drwx------ 3 root root  4096 Nov 30 23:12 systemd-private-85cdc6ab770f4767a4ebe49a8a6efda6-systemd-t
imesyncd.service-s6g7tg
-rw-rw-r-- 1 seed seed 15843 Nov 30 23:16 task3.zip
seed@recovery:/tmp$ unzip task3.zip
Archive:  task3.zip
   creating: task3/
   creating: task3/x86/
  inflating: task3/x86/mydaemon
  inflating: task3/x86/mysu
   creating: task3/META-INF/
   creating: task3/META-INF/com/
   creating: task3/META-INF/com/google/
   creating: task3/META-INF/com/google/android/
  inflating: task3/META-INF/com/google/android/app_process64
  inflating: task3/META-INF/com/google/android/update-binary
  inflating: task3/META-INF/com/google/android/mysu
seed@recovery:/tmp$ cd task3/
META-INF/ x86/
seed@recovery:/tmp$ cd task3/
META-INF/ x86/
seed@recovery:/tmp$ cd task3/META-INF/com/google/android/
seed@recovery:/tmp/task3/META-INF/com/google/android$ sudo ./update-binary
[sudo] password for seed:
```

```
x86_64:/ $ mysu
WARNING: linker: /system/xbin/mysu has text relocations. This is wasting memory and p
revents security hardening. Please fix.
start to connect to daemon
sending file descriptor
STDIN 0
STDOUT 1
STDERR 2
2
/system/bin/sh: No controlling tty: open /dev/tty: No such device or address
/system/bin/sh: warning: won't have full job control
x86_64:/ # cd /pr
proc/                property_contexts
x86_64:/ # ps | grep mysu
u0_a36    3295  3072  5064   1784           0 0000000000 S mysu
x86_64:/ # cd /proc/3295/fd
x86_64:/proc/3295/fd # ls -l
total 0
__bionic_open_tzdata_path: ANDROID_DATA not set!
__bionic_open_tzdata_path: ANDROID_ROOT not set!
lrwx------ 1 u0_a36 u0_a36 64 2018-12-01 04:21 0 -> /dev/pts/0
lrwx------ 1 u0_a36 u0_a36 64 2018-12-01 04:21 1 -> /dev/pts/0
lrwx------ 1 u0_a36 u0_a36 64 2018-12-01 04:21 2 -> /dev/pts/0
lrwx------ 1 u0_a36 u0_a36 64 2018-12-01 04:20 3 -> socket:[20227]
x86_64:/proc/3295/fd # ps | grep app_process
root      1060  1041  5064   292            0 0000000000 S /system/bin/app_process64
```

```
u0_a25    3245  1041  1112236 73700          0 0000000000 S com.android.vending:insta
nt_app_installer
u0_a36    3295  3072  5064   576            0 0000000000 S mysu
root      3296  1060  8316   2264           0 0000000000 S /system/bin/sh
u0_a17    3299  1041  1354016 137564          0 0000000000 S com.google.android.gms
u0_a17    3324  1041  1209852 129928          0 0000000000 S com.google.android.gms.p
ersistent
u0_a70    3558  1041  1950624 116984          0 0000000000 S com.google.android.youtu
be
u0_a17    3693  1041  1150240 89600          0 0000000000 S com.google.android.gms.ui
root      3948  3296  9880   2668           0 0000000000 R ps
x86_64:/proc/3295/fd # cd ../../3296/fd
x86_64:/proc/3296/fd # ls -l
total 0
__bionic_open_tzdata_path: ANDROID_DATA not set!
__bionic_open_tzdata_path: ANDROID_ROOT not set!
lrwx------ 1 root root 64 2018-12-01 04:24 0 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 1 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:20 10 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 2 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 4 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 5 -> socket:[5941]
lrwx------ 1 root root 64 2018-12-01 04:24 6 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 7 -> /dev/pts/0
lrwx------ 1 root root 64 2018-12-01 04:24 9 -> socket:[5282]
x86_64:/proc/3296/fd #
```

Questions:

Server launches the original app process binary:

File : mydaemonsu.c

Function: main()

Line no.: 255

Client sends its FDs

File : mysu.c

Function: connect_daemon()

Line no.: 112-114

Server forks to a child process

File : mydaemonsu.c

Function: main()

Line no.: 247

Child process receives client's FDs

File : mydaemonsu.c

Function: child_process()

Line no.: 147-149

Child process redirects its standard I/O FDs

File : mydaemonsu.c

Function: child_process()

Line no.: 152-154

Child process launches a root shell

File : mydaemonsu.c

Function: child_process()

Line no.: 162