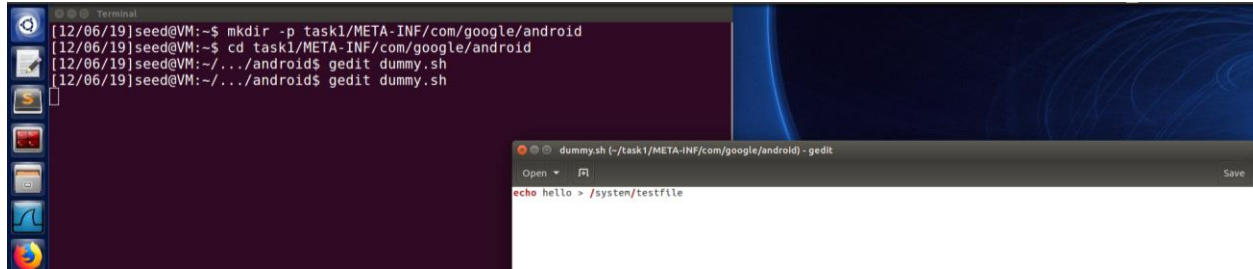


## Rooting Lab

### Task 1

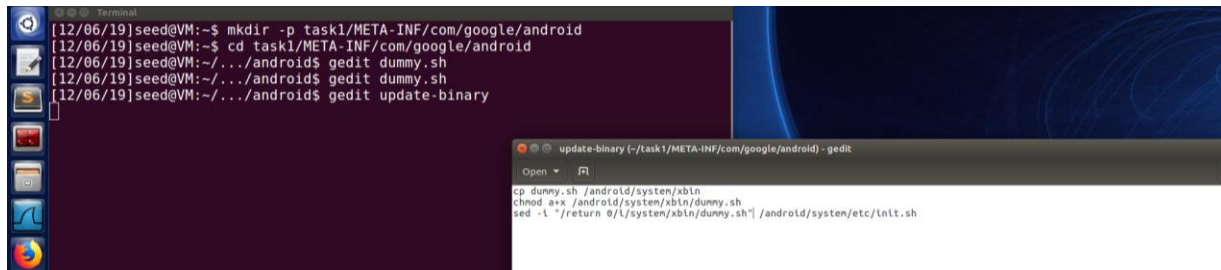


The terminal window shows the following commands and output:

```
[12/06/19]seed@VM:~$ mkdir -p task1/META-INF/com/google/android
[12/06/19]seed@VM:~$ cd task1/META-INF/com/google/android
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
```

The gedit window, titled "dummy.sh (~/.task1/META-INF/com/google/android) - gedit", contains the following text:

```
echo hello > /system/testfile
```

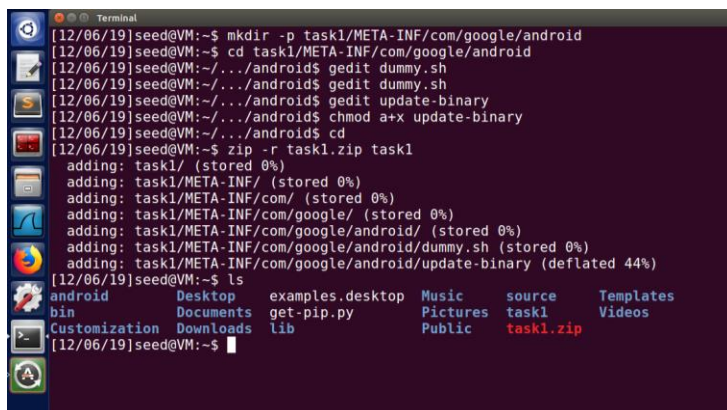


The terminal window shows the following commands and output:

```
[12/06/19]seed@VM:~$ mkdir -p task1/META-INF/com/google/android
[12/06/19]seed@VM:~$ cd task1/META-INF/com/google/android
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
[12/06/19]seed@VM:~/.../android$ gedit update-binary
```

The gedit window, titled "update-binary (~/.task1/META-INF/com/google/android) - gedit", contains the following text:

```
cp dummy.sh /android/system/sbin
chmod a+x /android/system/sbin/dummy.sh
sed -i "s/return 0;/return 0;/android/system/sbin/dummy.sh" /android/system/etc/init.sh
```



The terminal window shows the following commands and output:

```
[12/06/19]seed@VM:~$ mkdir -p task1/META-INF/com/google/android
[12/06/19]seed@VM:~$ cd task1/META-INF/com/google/android
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
[12/06/19]seed@VM:~/.../android$ gedit dummy.sh
[12/06/19]seed@VM:~/.../android$ gedit update-binary
[12/06/19]seed@VM:~/.../android$ chmod a+x update-binary
[12/06/19]seed@VM:~/.../android$ cd
[12/06/19]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%)
[12/06/19]seed@VM:~$ ls
android  Desktop  examples.desktop  Music  source  Templates
bin      Documents  get-pip.py        Pictures  task1  Videos
Customization  Downloads  lib              Public  task1.zip
```

```

Ubuntu 16.04.4 LTS recovery tty1

recovery login: seed
Password:
Last login: Fri May 18 15:17:56 EDT 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
seed@recovery:~$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:b9:66:9d
            inet addr:10.0.2.23  Bcast:10.0.2.255  Mask:255.255.255.0
            inet6 addr: fe80::a00:27ff:feb9:669d/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:19 errors:0 dropped:0 overruns:0 frame:0
            TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7095 (7.0 KB)  TX bytes:2808 (2.8 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:~$

```

```

[12/06/19]seed@VM:~$ ls
android  Desktop  examples.desktop  Music  source  Templates
bin      Documents  get-pip.py        Pictures  task1  Videos
Customization  Downloads  lib              Public  task1.zip

[12/06/19]seed@VM:~$ ping 10.0.2.23
PING 10.0.2.23 (10.0.2.23) 56(84) bytes of data:
64 bytes from 10.0.2.23: icmp_seq=1 ttl=64 time=0.969 ms
64 bytes from 10.0.2.23: icmp_seq=2 ttl=64 time=0.878 ms
^C
--- 10.0.2.23 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.878/0.923/0.969/0.054 ms
[12/06/19]seed@VM:~$ scp task1.zip seed@10.0.2.23:/tmp
The authenticity of host '10.0.2.23 (10.0.2.23)' can't be established.
ECDSA key fingerprint is SHA256:j27XN+nmbyA0avocRLHpOPiGRIZknAWmJli5y06vrsA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.23' (ECDSA) to the list of known hosts.
seed@10.0.2.23's password:
task1.zip                                100% 1406    1.4KB/s   00:00
[12/06/19]seed@VM:~$

```

```

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

seed@recovery:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:b9:66:9d
          inet addr:10.0.2.23  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb9:669d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:590 (590.0 B)  TX bytes:920 (920.0 B)

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 /tmp
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$ _

```

```

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

```

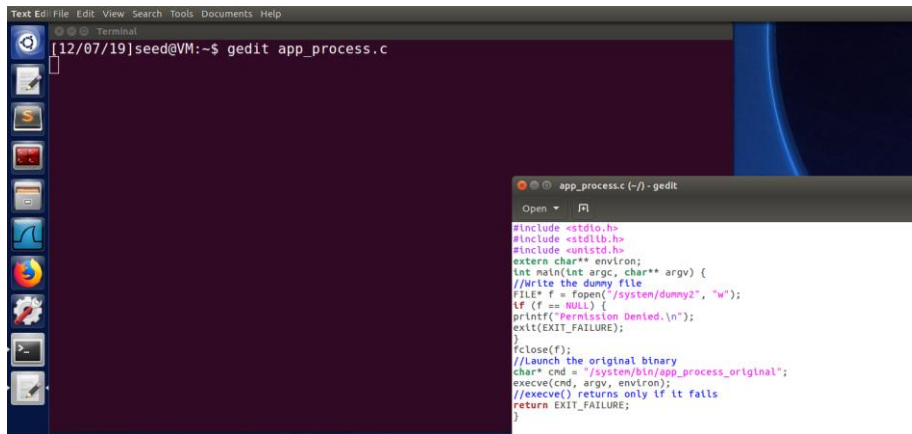
```

x86_64:/ $ cd /s
sbin/                selinux_version     storage/
sdcard/              sepolicy             sys/
seapp_contexts       service_contexts    system/
x86_64:/ $ cd /sys
sys/                 system/
x86_64:/ $ cd /system/
x86_64:/system $ ls
app  build.prop  fake-libs  fonts  lib  lost+found  priv-app  usr  xbin
bin  etc        fake-libs64  framework  lib64  media  testfile  vendor
x86_64:/system $ █

```

In this task we added some content in the dummy.sh script that we created. After which we also added content to the update-binary file to copy the dummy.sh file to the system folder so that as the OS recovery starts it opens up our file

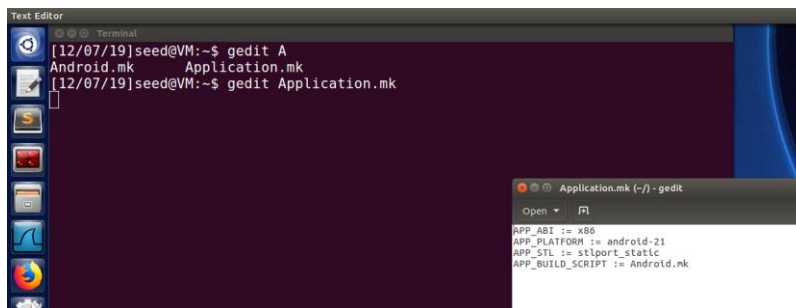
## Task 2



```
Text Editor
[12/07/19]seed@VM:~$ gedit app_process.c

app_process.c (~/) - gedit
Open  [F]

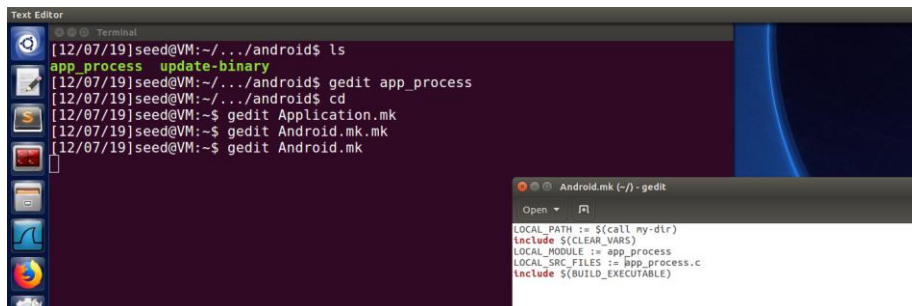
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
extern char** environ;
int main(int argc, char** argv) {
    //Write the dummy file
    FILE* f = fopen("/system/dummy2", "w");
    if (f == NULL) {
        printf("Permission Denied.\n");
        exit(EXIT_FAILURE);
    }
    fclose(f);
   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



```
Text Editor
[12/07/19]seed@VM:~$ gedit A
Android.mk
[12/07/19]seed@VM:~$ gedit Application.mk

Application.mk (~/) - gedit
Open  [F]

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



```
Text Editor
[12/07/19]seed@VM:~/../android$ ls
app_process  update-binary
[12/07/19]seed@VM:~/../android$ gedit app_process
[12/07/19]seed@VM:~/../android$ cd
[12/07/19]seed@VM:~$ gedit Application.mk
[12/07/19]seed@VM:~$ gedit Android.mk.mk
[12/07/19]seed@VM:~$ gedit Android.mk

Android.mk (~/) - gedit
Open  [F]

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

```
Text Editor
Terminal
[12/07/19]seed@VM:~$ gedit app_process.c
[12/07/19]seed@VM:~$ cp app_process task2/META-INF/com/google/android/
[12/07/19]seed@VM:~$ cd task2/META-INF/com/google/android/
[12/07/19]seed@VM:~/.../android$ ls
app_process  update-binary
[12/07/19]seed@VM:~/.../android$ chmod a+x app_process
[12/07/19]seed@VM:~/.../android$ ls
app_process  update-binary
[12/07/19]seed@VM:~/.../android$ gedit update-binary
update-binary (~/.task2/META-INF/com/google/android) - gedit
Open
mv /android/system/bin/app_process64 /android/system/bin/app_process_original
cp app_process /android/system/bin/app_process64
chmod a+x /android/system/bin/app_process64
chmod a+x update-binary
```

```
Terminal
[12/07/19]seed@VM:~/.../android$ cd
[12/07/19]seed@VM:~$ zip -r task2.zip task2
updating: task2/ (stored 0%)
updating: task2/META-INF/ (stored 0%)
updating: task2/META-INF/com/ (stored 0%)
updating: task2/META-INF/com/google/ (stored 0%)
updating: task2/META-INF/com/google/android/ (stored 0%)
updating: task2/META-INF/com/google/android/update-binary (deflated 57%)
updating: task2/META-INF/com/google/android/app_process (deflated 64%)
[12/07/19]seed@VM:~$
```

```

seed@recovery:~$ cd /tmp/
seed@recovery:/tmp$ ls
systemd-private-8202b7e6dc104b2db70545812089f1dc-systemd-timesyncd.service-WucSLC
seed@recovery:/tmp$ ls
systemd-private-8202b7e6dc104b2db70545812089f1dc-systemd-timesyncd.service-WucSLC
seed@recovery:/tmp$ ls
systemd-private-8202b7e6dc104b2db70545812089f1dc-systemd-timesyncd.service-WucSLC 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/update-binary
  inflating: task2/META-INF/com/google/android/app_process
seed@recovery:/tmp$ cd task2/META-INF/com/google/android/
seed@recovery:/tmp/task2/META-INF/com/google/android$ ls
app_process  update-binary
seed@recovery:/tmp/task2/META-INF/com/google/android$ sudo ./update-binary
[sudo] password for seed:
seed@recovery:/tmp/task2/META-INF/com/google/android$ sudo reboot

```

```

x86_64:/ $ cd /system
x86_64:/system $ ls
app          dummy2      fake-libs64  lib          media        usr
bin          etc         fonts        lib64        priv-app     vendor
build.prop  fake-libs   framework    lost+found   testfile     xbin
x86_64:/system $ █

```

In this task as seen from above screenshots we first created a directory named task2 in which we added files – app\_process.c, application.mk and android.mk. After which we compile it. And if it succeeds we can find the binary file ./libs/x86 folder .

Here, we also wrote code for update-binary . Later, we zip the task2 OTA package and send to OS recovery. In the OS recovery we unzip the package and run. If we are successful we the android is booted and after checking the /system folder we can see the dummy2 file that we wrote in the app\_process file.

### Task 3

```
Terminal
[12/06/19]seed@VM:~$ mkdir -p task3/META-INF/com/google/android/
[12/06/19]seed@VM:~$ cd Downloads/
[12/06/19]seed@VM:~/Downloads$ unzip SimpleSU.zip
Archive: SimpleSU.zip
  creating: SimpleSU/
  creating: SimpleSU/socket_util/
  inflating: SimpleSU/socket_util/socket_util.c
  inflating: SimpleSU/socket_util/socket_util.h
  creating: SimpleSU/mydaemon/
  inflating: SimpleSU/mydaemon/Android.mk
  inflating: SimpleSU/mydaemon/compile.sh
  inflating: SimpleSU/mydaemon/mydaemonsu.c
  inflating: SimpleSU/mydaemon/Application.mk
  inflating: SimpleSU/compile_all.sh
  inflating: SimpleSU/server_loc.h
  creating: SimpleSU/mysu/
  inflating: SimpleSU/mysu/Android.mk
  inflating: SimpleSU/mysu/compile.sh
  inflating: SimpleSU/mysu/mysu.c
  inflating: SimpleSU/mysu/Application.mk
[12/06/19]seed@VM:~/Downloads$
```

```
Terminal
[12/06/19]seed@VM:~/Downloads$ cd SimpleSU/
[12/06/19]seed@VM:~/../SimpleSU$ bash compile_all.sh
////////Build Start////////
Compile x86 : mydaemon <= mydaemonsu.c
Compile x86 : mydaemon <= socket_util.c
Executable : mydaemon
Install : mydaemon => libs/x86/mydaemon
Compile x86 : mysu <= mysu.c
Compile x86 : mysu <= socket_util.c
Executable : mysu
Install : mysu => libs/x86/mysu
////////Build End////////
[12/06/19]seed@VM:~/../SimpleSU$ bash compile_all.sh
////////Build Start////////
Install : mydaemon => libs/x86/mydaemon
Install : mysu => libs/x86/mysu
////////Build End////////
[12/06/19]seed@VM:~/../SimpleSU$
```

```
Terminal
[12/07/19]seed@VM:~$ ls
android  app_process Customization Downloads lib my_app_process.c Public task1.zip task3 Videos
Android.mk app_process.c Desktop examples.desktop libs obj
Application.mk bin Documents get-pip.py Music Pictures source task2 task3.zip
[12/07/19]seed@VM:~$ cd task3
[12/07/19]seed@VM:~/task3$ ls
META-INF x86
[12/07/19]seed@VM:~/task3$ cd x86/
[12/07/19]seed@VM:~/../x86$ ls
mydaemon mysu
[12/07/19]seed@VM:~/../x86$ cd ..
[12/07/19]seed@VM:~/task3$ ls
META-INF x86
[12/07/19]seed@VM:~/task3$ cd META-INF/
[12/07/19]seed@VM:~/../META-INF$ ls
com
[12/07/19]seed@VM:~/../META-INF$ cd com/google/android/
[12/07/19]seed@VM:~/../android$ ls
update-binary
[12/07/19]seed@VM:~/../android$
```

```
File Edit View Search Tools Documents Help
Open
mv /android/system/bin/app_process64 /android/system/bin/app_process_original
cp ../x86/mydaemon /android/system/bin/app_process64
cp ../x86/mysu /android/system/bin/mysu
chmod a+x /android/system/bin/app_process64
chmod a+x /android/system/bin/mysu
```

```
Terminal
[12/07/19]seed@VM:~/.../android$ ls
[12/07/19]seed@VM:~/.../android$ gedit update-binary
[12/07/19]seed@VM:~/.../android$ chmod a+x update-binary
[12/07/19]seed@VM:~/.../android$ cd
[12/07/19]seed@VM:~$ zip task3.zip task3
  adding: task3/ (stored 0%)
[12/07/19]seed@VM:~$ zip -r task3.zip task3
  updating: task3/ (stored 0%)
  adding: task3/x86/ (stored 0%)
  adding: task3/x86/mydaemon (deflated 60%)
  adding: task3/x86/mysu (deflated 66%)
  adding: task3/META-INF/ (stored 0%)
  adding: task3/META-INF/com/ (stored 0%)
  adding: task3/META-INF/com/google/ (stored 0%)
  adding: task3/META-INF/com/google/android/ (stored 0%)
  adding: task3/META-INF/com/google/android/update-binary (deflated 63%)
[12/07/19]seed@VM:~$ scp task3.zip seed@10.0.2.27:/tmp
The authenticity of host '10.0.2.27 (10.0.2.27)' can't be established.
ECDSA key fingerprint is SHA256:j27XM-nmbyA0avocrLHpQPjGRIZknAWmJli5y06vrsA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.27' (ECDSA) to the list of known hosts.
seed@10.0.2.27's password:
task3.zip                                100% 8542    8.3KB/s   00:00
```

```
seed@recovery:/tmp$ ls
systemd-private-d6d1c9cc83a741818649a740cdc9c3e8-systemd-timesyncd.service-cCoue2 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/update-binary
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:
seed@recovery:/tmp/task3/META-INF/com/google/android$ sudo reboot
```



```

x86_64:/ $ id
uid=10036(u0_a36) gid=10036(u0_a36) groups=10036(u0_a36),3003(inet),9997(everybody),5
0036(all_a36) context=u:r:untrusted_app:s0:c512,c768
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:/ # █

```

In this task we first unzip the SimpleSU file. Later we create x86 folder and copy mydaemon and mysu file. Now we create a new update-binary file. And then after zipping everything we send it to OS recovery and then run the update binary from there and then reboot android vm. We can see that we are successful in getting the root shell.

1. Server Launches the original app process binary :
  - a. Filename : mydaemonsu.c
  - b. Function: main()
  - c. Line : 255
2. Client sends FDs:
  - a. Filename: mysu.c
  - b. Function: connect\_deamon()
  - c. Line: 112 to 114
3. Server forks to a child process in :
  - a. Filename: mydaemonsu.c
  - b. Function: main()
  - c. Line: 247
4. Child process receives clients FD:
  - a. Filename: mydeamonsu.c
  - b. Function: child\_process()
  - c. Line: 147 to 149
5. Child process redirects its standard i/o FDs
  - a. Filename: mydeamonsu.c
  - b. Function: child\_process()
  - c. Line: 152 to 154
6. Child process launches root shell
  - a. Filename: mysu.c
  - b. Function: main()
  - c. Line: 154

