

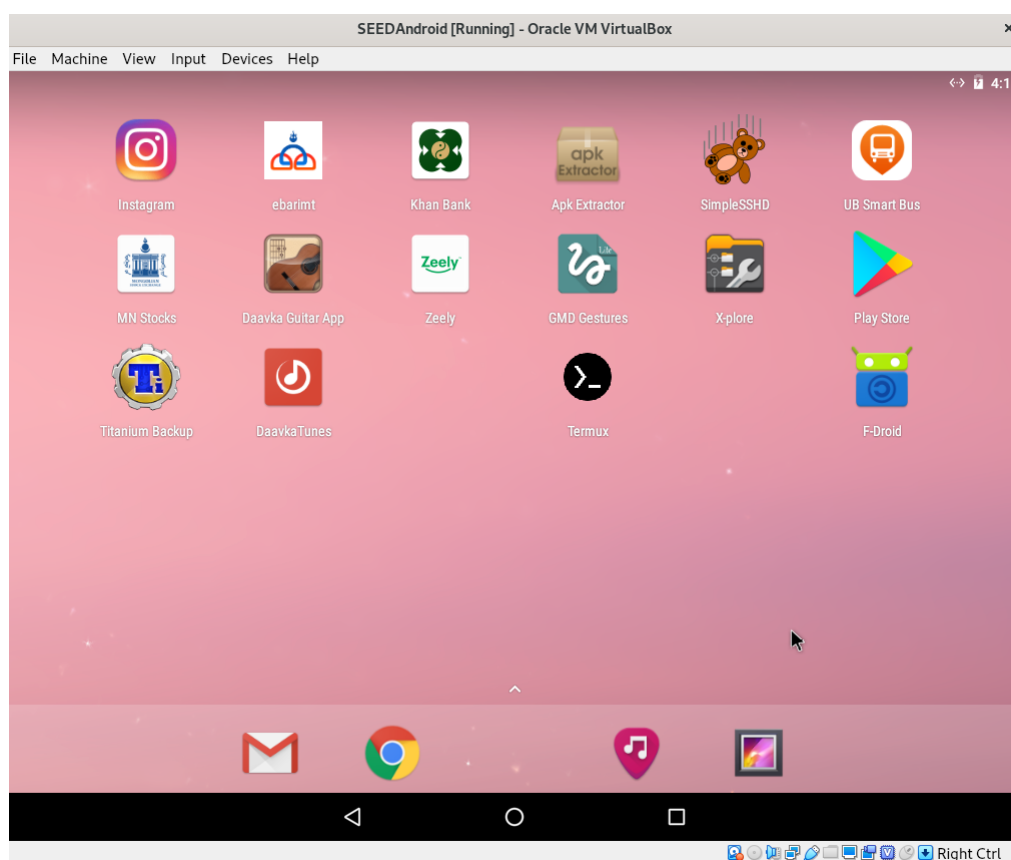
1 Хураангуй

Андройд төхөөрөмжийг *root* хийх (*rooting*) үйл явцтай танилцах ба үүнийг хийхийн тулд ямар алхмууд хэрэгтэй вэ гэдгийг ойлгох зорилготой. Мөн, *rooting* механизм нь андройд систем болон ерөнхийдөө үйлдлийн системийн талаарх өргөн мэдлэг чадварыг хамардаг. Энэ нь системийн гүнзгий мэдлэгийг олж авах маш сайн хэрэгсэл (*vehicle*) болж өгдөг. Энэ лабораториор *rooting* багцыг хөгжүүлж, түүнийгээ ашиглан Android VM -ыг *rooting* хийнэ.

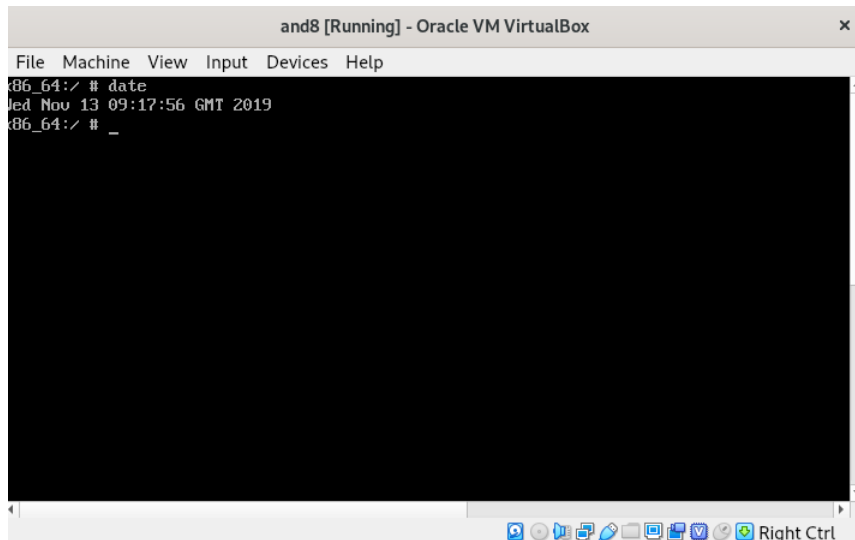
2 Хэрэгжүүлэлт

2.1 Лабораторийн орчин бэлдэх

Android VM x86 Nougat



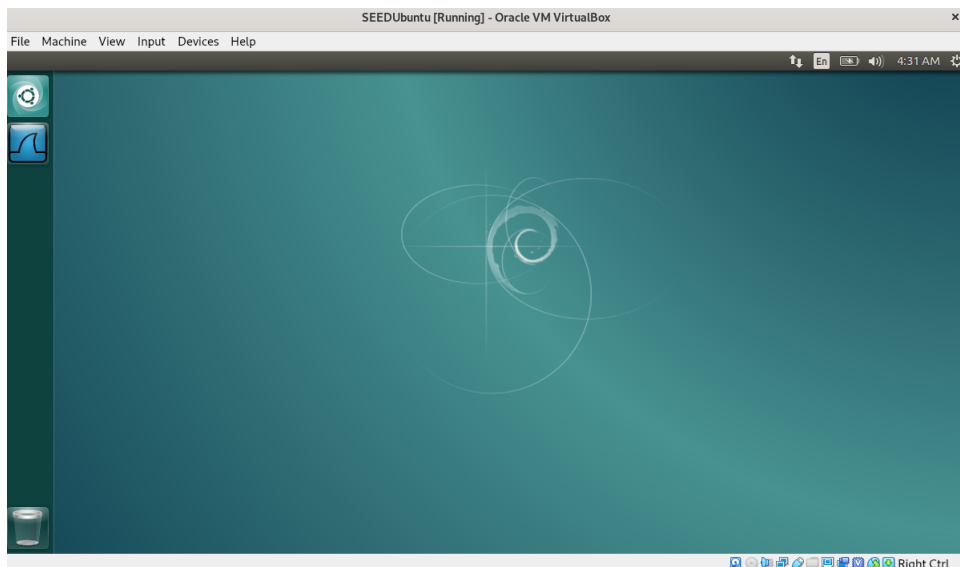
Зураг 1: Android VM (GUI)



Зураг 2: Android VM (CLI)

Recovery OS

Үнэндээ, бид “Ubuntu 16.04”-ыг “*recoveryOS*” болгон ашиглаж байгаа.



Зураг 3: Ubuntu 16.04 (recoveryOS)

2.2 Task 1: Build a simple OTA package

Энэ даалгавраар энгийн *OTA* пакетыг оргүй хоосноос бүтээх (build). үйлдлийн системийг *root* эрхээр ашиглахын тулд энэ пакет хэрэгтэй юм.

Үүний тулд жижиг зорилгууд тавьж ажиллах хэрэгтэй.

- *RecoveryOS* -оос андройд үйлдлийн систем рүү програм хэрхэн тарих (inject) талаар мэдэх?
- Тарьсан программаа хэрхэн *root* эрхээр автоматаар ажиллуулах?
- shell -д *root* эрхээр орхоос илүү программ хэрхэн бичих талаар?

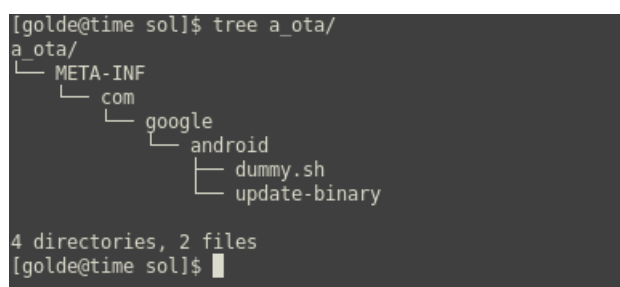
dummy.sh дотор дараах тушаал байна.

```
1 echo hello > /system/dummy
```

Алхам 1. *update* скрипт бичих

update – binary нь 2 хэсэгтэй:

1. Андройд үйлдлийн систем рүү *dummy.sh* програмыг тарих (inject): *RecoveryOS* дахь */android* хавтастай *Android* үйлдлийн системийн партешейнг mount хийх хэрэгтэй.
2. Андройд үйлдлийн системийн тохиргооны файлыг өөрчлөх: *dummy.sh* андройд бүүтлэг-лэгдэх үед *root* эрхээр автоматаар ажиллах боломжтой.



```
[golde@time sol]$ tree a_ota/
a_ota/
├── META-INF
│   └── com
│       └── google
│           └── android
│               ├── dummy.sh
│               └── update-binary
└── 4 directories, 2 files
[golde@time sol]$
```

Зураг 4: OTA бүтэц

update – binary дотор дараах скрипт байна.

```
1 cp dummy.sh /android/system/xbin
2 chmod a+x /android/system/xbin/dummy.sh
3 sed -i /return 0/i/system/xbin/dummy.sh /android/system/etc/init.sh
```

Алхам 2. *OTA* пакет үүсгэх Дээрх *ota* бүтцээр үүсгэсэн файл болон скриптээ *zip* хийх замаар *OTA* пакетаа үүсгэнэ. Дараах тушаалын дагуу үүсгэнэ.

```
1 zip -r my_ota.zip ./
```

Харин, доор тушаалын дагуу дотор нь юу агуулж буйг харж болно.

```
1 unzip -l my_ota.zip
```

Алхам 3. OTA пакетыг ажиллуулах

RecoveryOS нь *recovery* функцуудтай байх ёстой. *OTA* пакетыг гараар задлах (*unzip* ашиглан) ёстой. META-INF/com/google/android доторх *update* – *binary* ажиллуулснаар андройд *update* хийгдлээ. Өөрөөр хэлбэл, андройд үйлдлийн систем бүүтлэгдэх үед “/system” дотор *dummy* файл үүсэх болно.

```
1 :/ # scp my_ota.zip seed192.168.1.5:/home
```

Дээрх тушаалын үр дүнд *RecoveryOS*-д *OTA* пакет хуулагдах юм.

```
1 seedVM: unzip my_ota.zip
```

Дээрх тушаалын үр дүнд пакет *RecoveryOS*-д задрах юм. Үүний дараа “*update* – *script*”-ыг ажиллуулна.

Үр дүнд нь:

```
init.baseband.rc      sepolicy
init.environ.rc       storage
init.power.rc         sys
init.rc               system
init.recovery.samsungexynos7580.rc ueventd.rc
init.samsung.rc       ueventd.samsungexynos7580.rc
init.samsungexynos7580.rc vendor
init.samsungexynos7580.usb.rc vendor_file_contexts
init.target.rc        vendor_hwservice_contexts
init.usb.configfs.rc  vendor_property_contexts
init.usb.rc           vendor_seapp_contexts
init.wifi.rc          vendor_service_contexts
init.zygote32.rc      vndservice_contexts
:/ # cd /system
:/system # ls
addon.d      compatibility_matrix.xml framework  priv-app      vendor
app          etc          lib        product      xbin
bin          fake-libs   lost+found recovery-from-boot.p
build.prop   fonts       media      usr
:/system # █
```

Зураг 5: Андройд үйлдлийн системийн *system* фолдерт хандсан

2.3 Task 2: Inject code via app process

Алхам 1. Кодоо компайлдах

my_app_process.c код доорхтой адил байна.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4
5 extern char** environ;
6
7 int main(int argc, char** argv) {
8     //Write the dummy file
9     // End double quotation mark bga bolno
10    FILE* f = fopen("/system/dummy2", w);
11
12    if (f == NULL) {
13        printf(Permission Denied.\n);
```

```

14     exit(EXIT_FAILURE);
15 }
16
17 fclose(f);
18
19 //Launch the original binary
20 char* cmd = /system/bin/app_process_original;
21 execve(cmd, argv, environ);
22
23 //execve() returns only if it fails
24 return EXIT_FAILURE;
25 }

```

Android.mk тохиргоо доорх хэлбэртэй бичэгдсэн.

```

1 LOCAL_PATH := $(call my-dir)
2 include $(CLEAR_VARS)
3 LOCAL_MODULE := my_app_process
4 LOCAL_SRC_FILES := my_app_process.c
5 include $(BUILD_EXECUTABLE)

```

Application.mk тохиргооны агуулга

```

1 APP_ABI := x86
2 APP_PLATFORM := android-22
3 APP_STL := stlport_static
4 APP_BUILD_SCRIPT := Android.mk

```

Native development kit-ээр *build* хийхдээ дараах байдлаар хийнэ.

```

1 export NDK_PROJECT_PATH=.
2 ndk-build NDK_APPLICATION_MK=./Application.mk

```

Алхам 2. *update* – *script* бичих ба *OTA* пакет үүсгэх
update – *binary* -ын агуулга

```

1 mv /android/system/bin/app_process32 /android/system/bin/app_process_original
2 cp my_app_process /android/system/bin/app_process32
3 chmod a+x /android/system/bin/app_process32

```

Task 1 дээр хийсэнтэй адил *OTA* пакет үүсгэнэ. Ингэхдээ, *compile.sh* -ыг ажиллуулна. Үүний үр дүнд *OTA* бүтэц бүхий файлд бидэнд хэрэгцээтэй ажиллахуйц программ болон шаардлагатай файлууд үүснэ үүнийг *zip* хийгээд *scp*-ээр *recoveryOS*-руу хуулж өгнө. Үүний дараа гар аргаар дахин *recovery* -гоос *unzip* хийж *update* – *script* -ийг ажиллуулна. Үүний үр дүнд:

```

u0_a27@x86:/ $ cd /system
u0_a27@x86:/system $ ls
app
bin
build.prop
dummy2
etc
fonts
framework
lib
lost+found
media
priv-app
testfile
usr
vendor
xbin
u0_a27@x86:/system $ █

```

Зураг 6: `/system` директорт `dummy2` үүсгэсэн байдал

2.4 Task 3: Implement SimpleSU for Getting Root Shell

`update – binary` -ын агуулга

```

1 cp mysu /android/system/xbin
2 cp mydaemon /android/system/xbin
3 sed -i /return 0/i /system/xbin/mydaemon /android/system/etc/init.sh

```

Өмнөх даалгаваруудад хийсэн шиг мөн л `update – script`-ээ бичсэн бол *OTA* пакетаа үүсгээд түүнийгээ *recoveryOS* -руугаа хуулаад, задлаад бэлднэ. Үүний дараа, *mysu*-г *recoveryOS*-оос ажиллуулна. Үүний үр дүнд:

```

$ su
:/ # id
uid=0(root) gid=0(root) groups=0(root) context=u:r:su:daemon:s0
:/ # █

```

Зураг 7: *root* эрх авсан байдал

3 Дүгнэлт

Энэ лаборатори нь андройд үйлдлийн системийг хэрхэн *rooting* хийх талаар нарийн ойлголтыг өгч байна. Андройд *rooting* хийхдээ бид *recoveryOS* -оос *OTA* пакетийг суулгаж түүнийг систем бүүтлэгдэх үед хамт ажиллуулах замаар админ эрх олж авлаа.

Мөн, андройд утас хэрэглэдэг бол утас *rooting* хийвэл ямар давуу болон сул талууд үүсч болох нь сайн харагдаж, мэдрэгдэж байна. Эндээс, *rooting* хийснээр сул тал хэдий байдаг ч гэсэн *rooting* хийсэн нь дээр гэж дүгнэж байна. Учир нь, ялангуяа хөгжүүлэгчид энэ эрхээр хандаж орох нэн шаардлагатай байдаг ба *root* эрхийн тусламжтай өөрийн хөгжүүлж буй ашпын аюулгүй байдлыг туршиж үзэхэд нэн хэрэгтэй болох нь харагдаж байна.

4 Ашигласан материал

Ашигласан ном

- [1] SEEDUbuntu: SEEDVM VirtualBox Manual,
https://seedsecuritylabs.org/Labs_16.04/Documents/SEEDVM_VirtualBoxManual.pdf
- [2] How to run the Android VM in VirtualBox
https://seedsecuritylabs.org/Labs_16.04/Mobile/SEEDAndroid_VirtualBox.pdf
- [3] User Manual for the Android OS
https://seedsecuritylabs.org/Labs_16.04/Mobile/SEEDAndroid_UserManual.pdf