

## 팀 과제 설명

2019.11.20

조재희

jehee1204@gmail.com

# 목차

---

- ❖ 환경 구성
- ❖ 배경 지식
- ❖ 실습 개요 / 방법
- ❖ 평가기준 / 제출

# 환경구성

2019.11.20

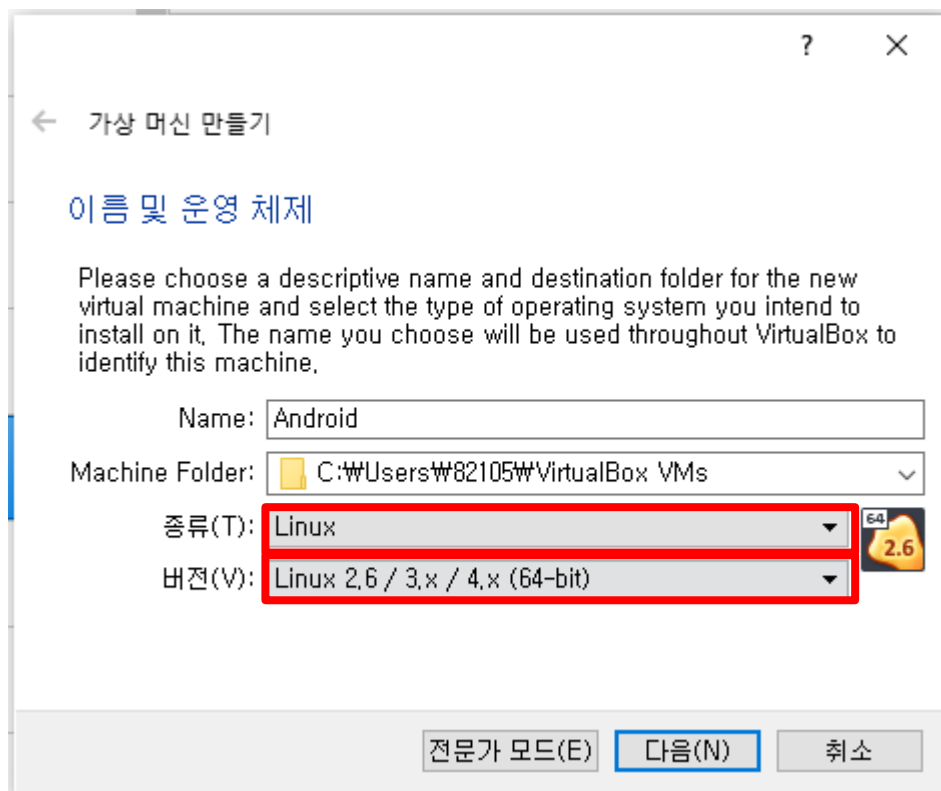
조재희

jehee1204@gmail.com

## ❖ Android VM 다운로드

- Android\_CSOS.vmdk
  - <http://seuresw.dankook.ac.kr>

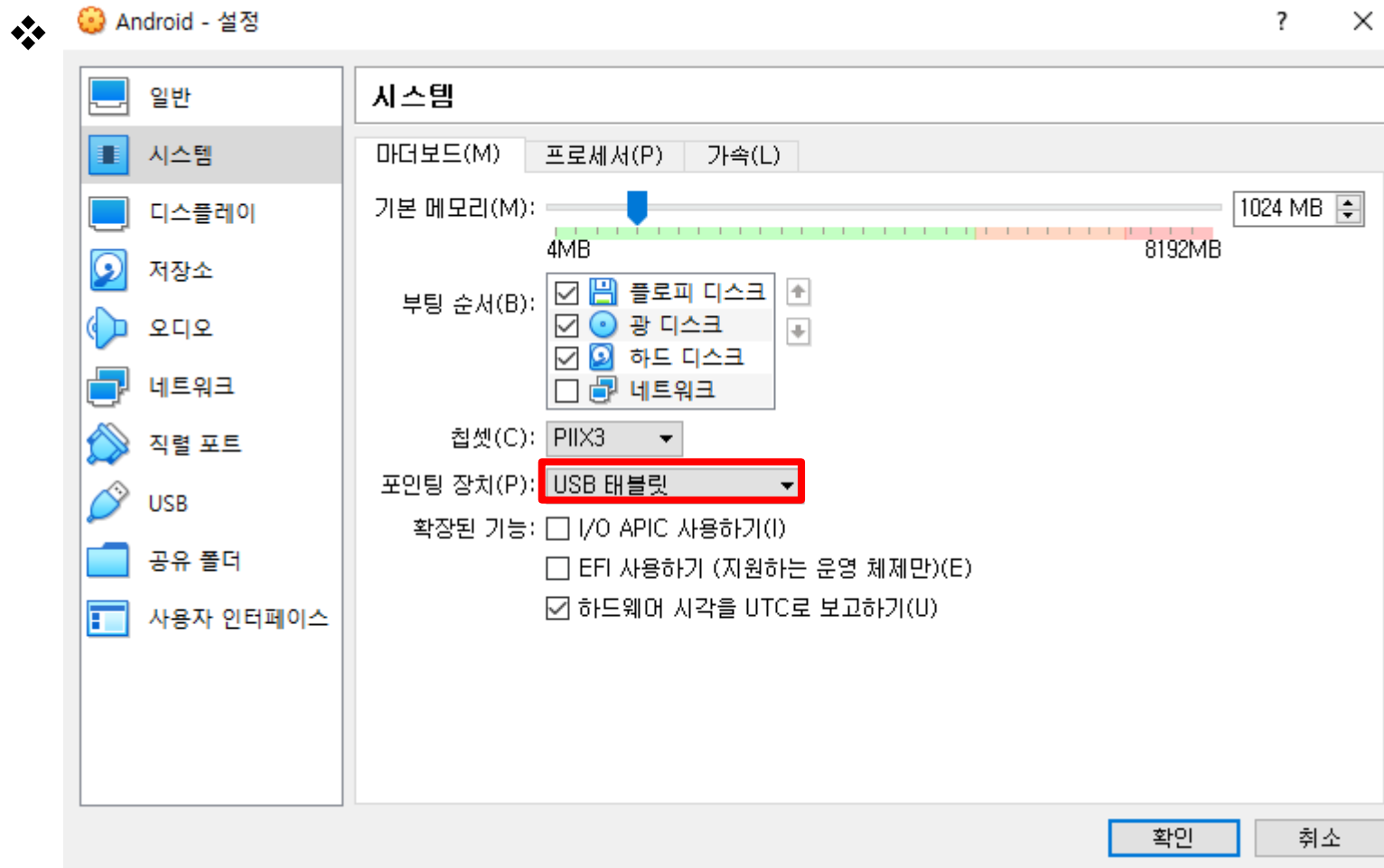
# 환경구성\_가상머신 만들기



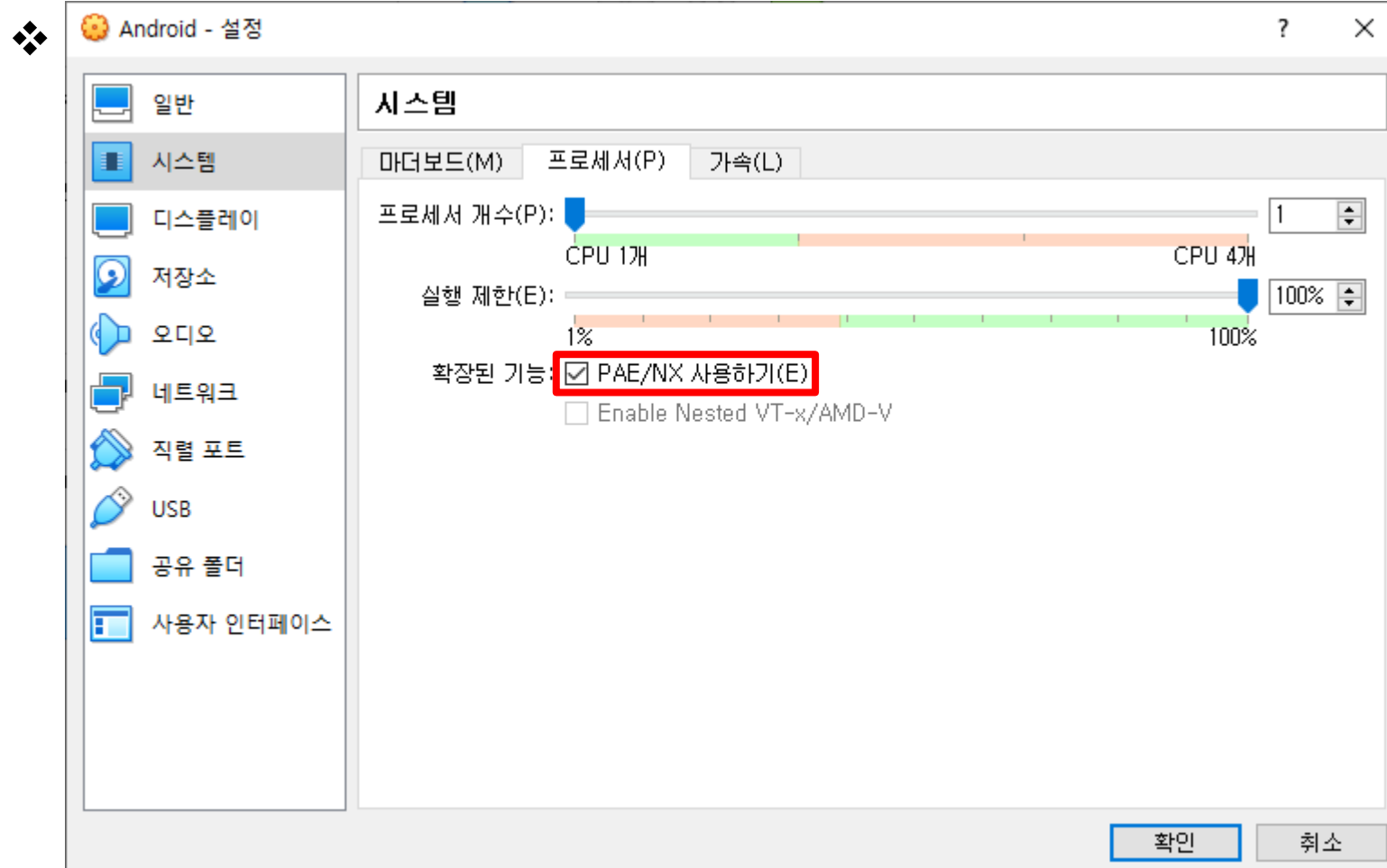
## ❖ 가상 머신 만들기

- 메모리 크리 1024 MB
- 기존 가상 하드 디스크 파일 사용
  - 홈페이지에서 다운받은 Android\_CSOS.vmdk 선택

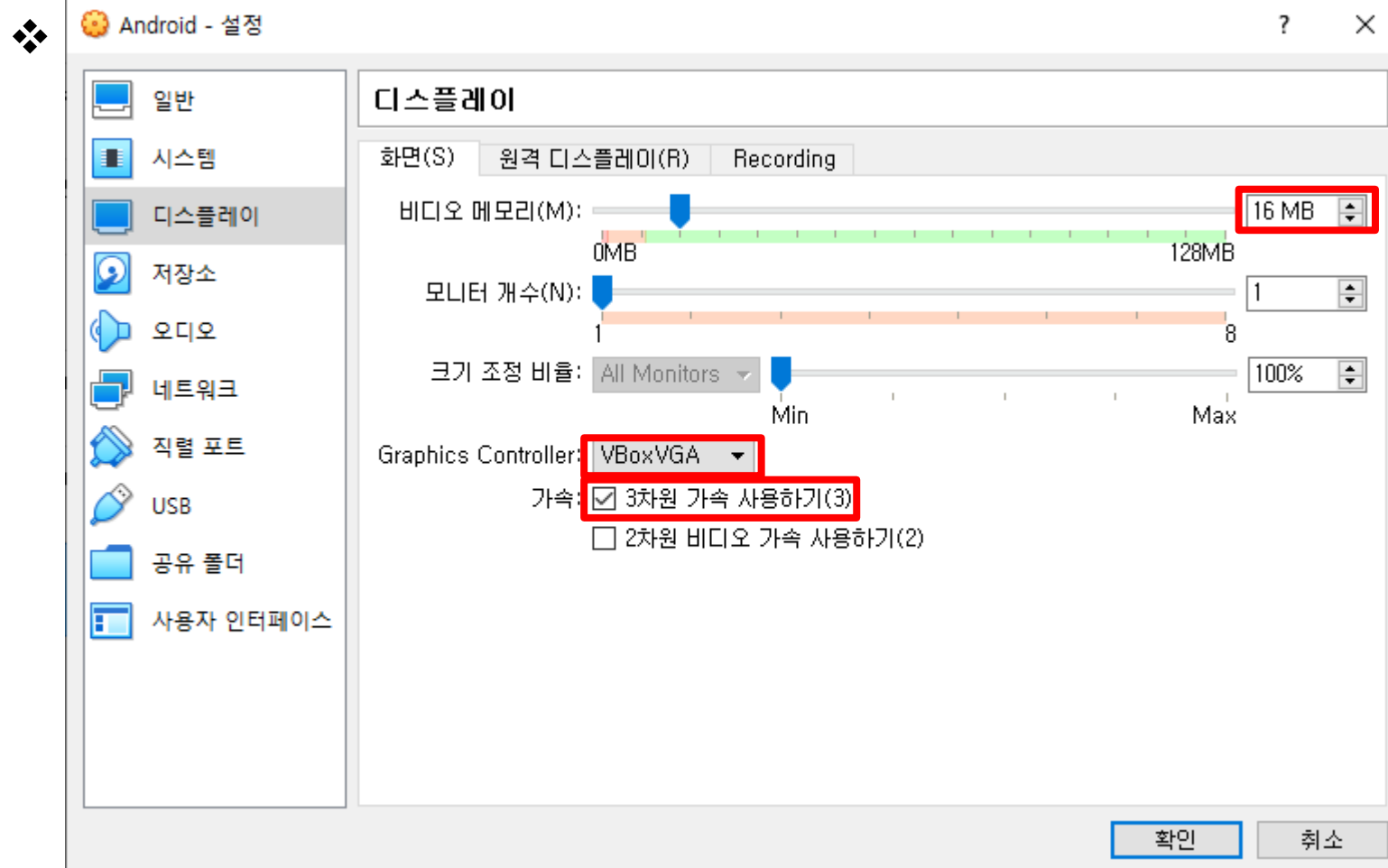
# 환경구성\_Android 설정



# 환경구성\_Android 설정



# 환경구성\_Android 설정

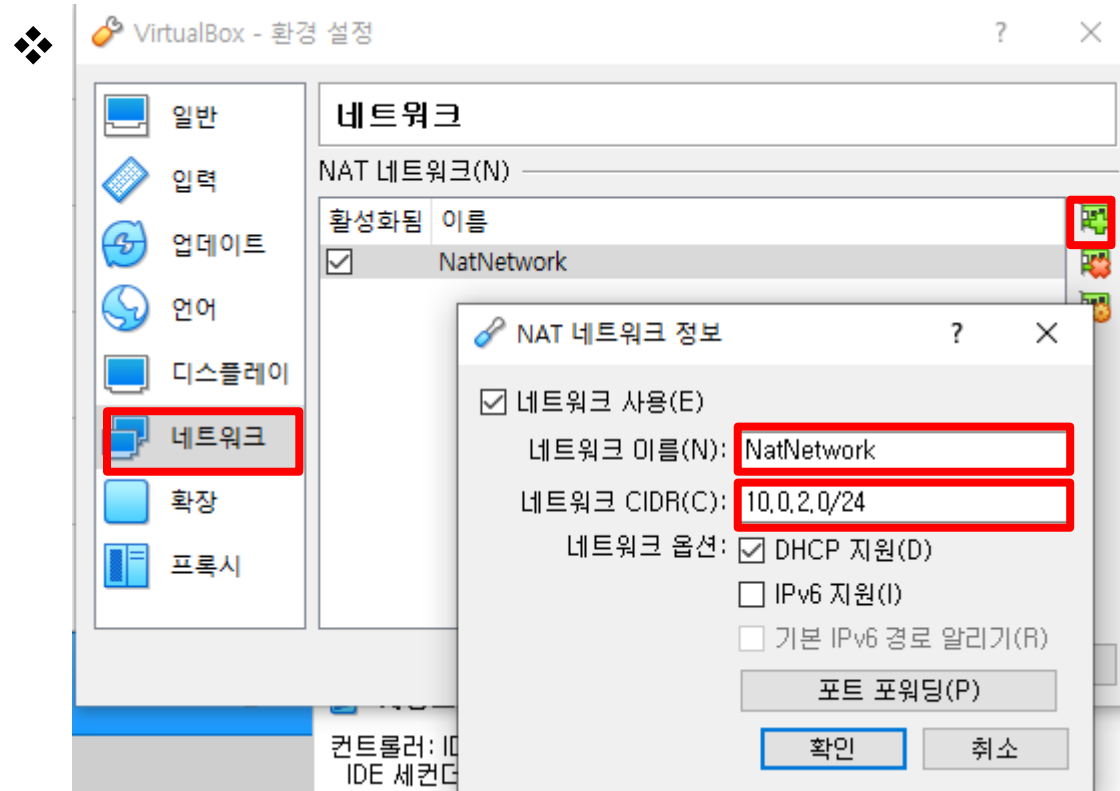




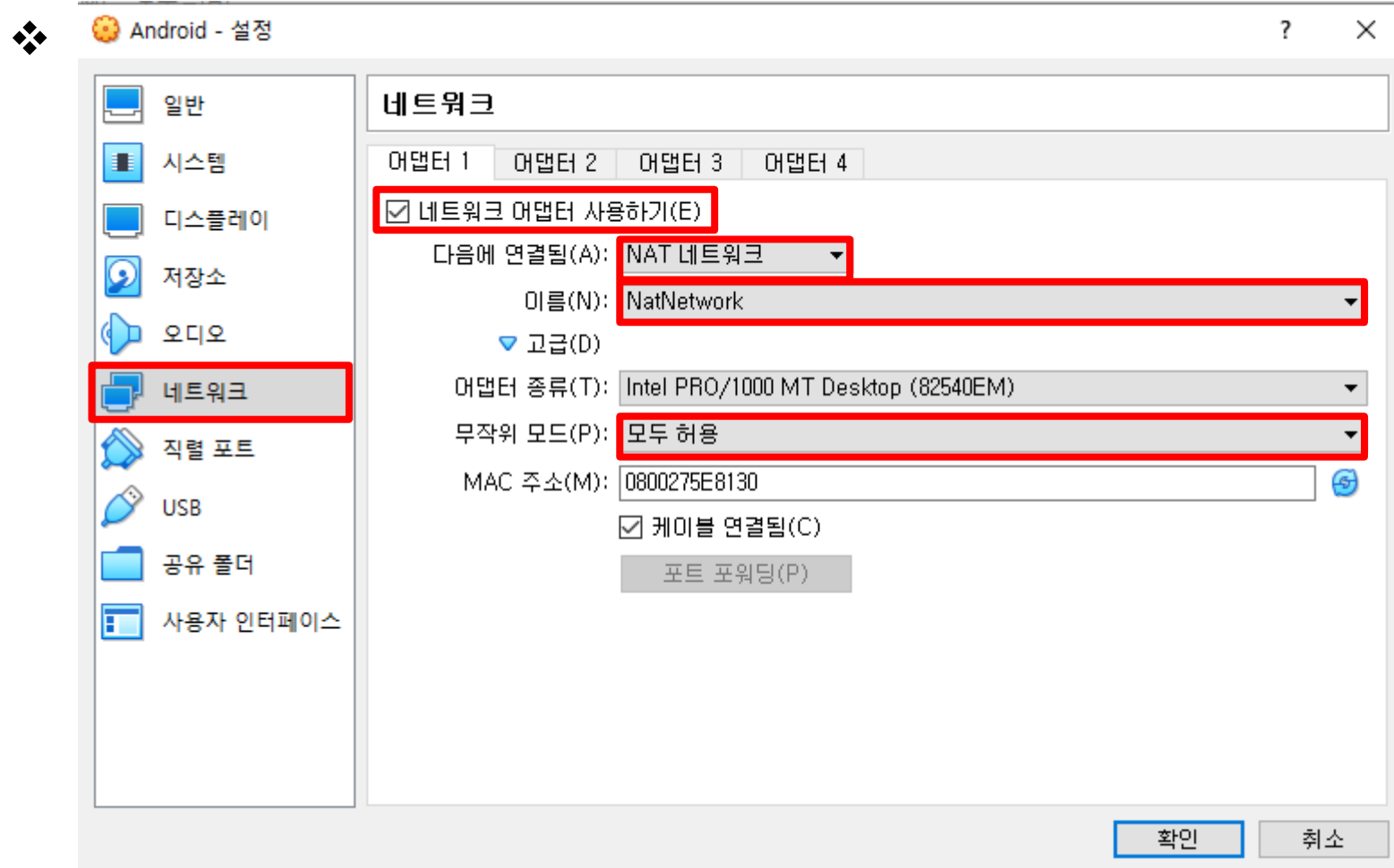
# 환경구성\_환경설정 (네트워크)



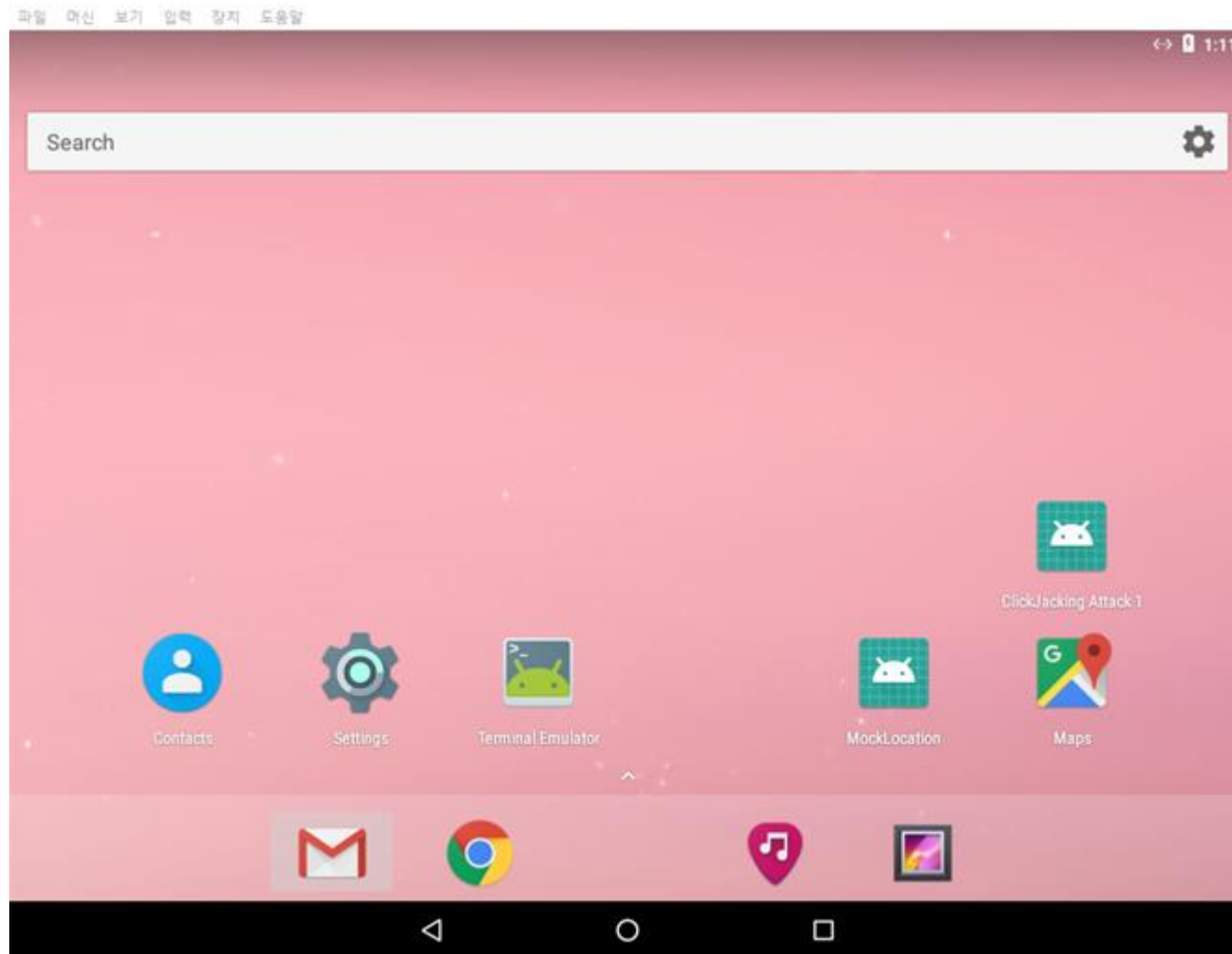
# 환경구성\_환경설정 (네트워크)



# 환경구성\_Android 설정



## ❖ 실행



# Android Rooting Attack

Android VM을 활용하여 rooting 실습

2019.11.20

조재희

jehee1204@gmail.com

# Android Rooting Attack

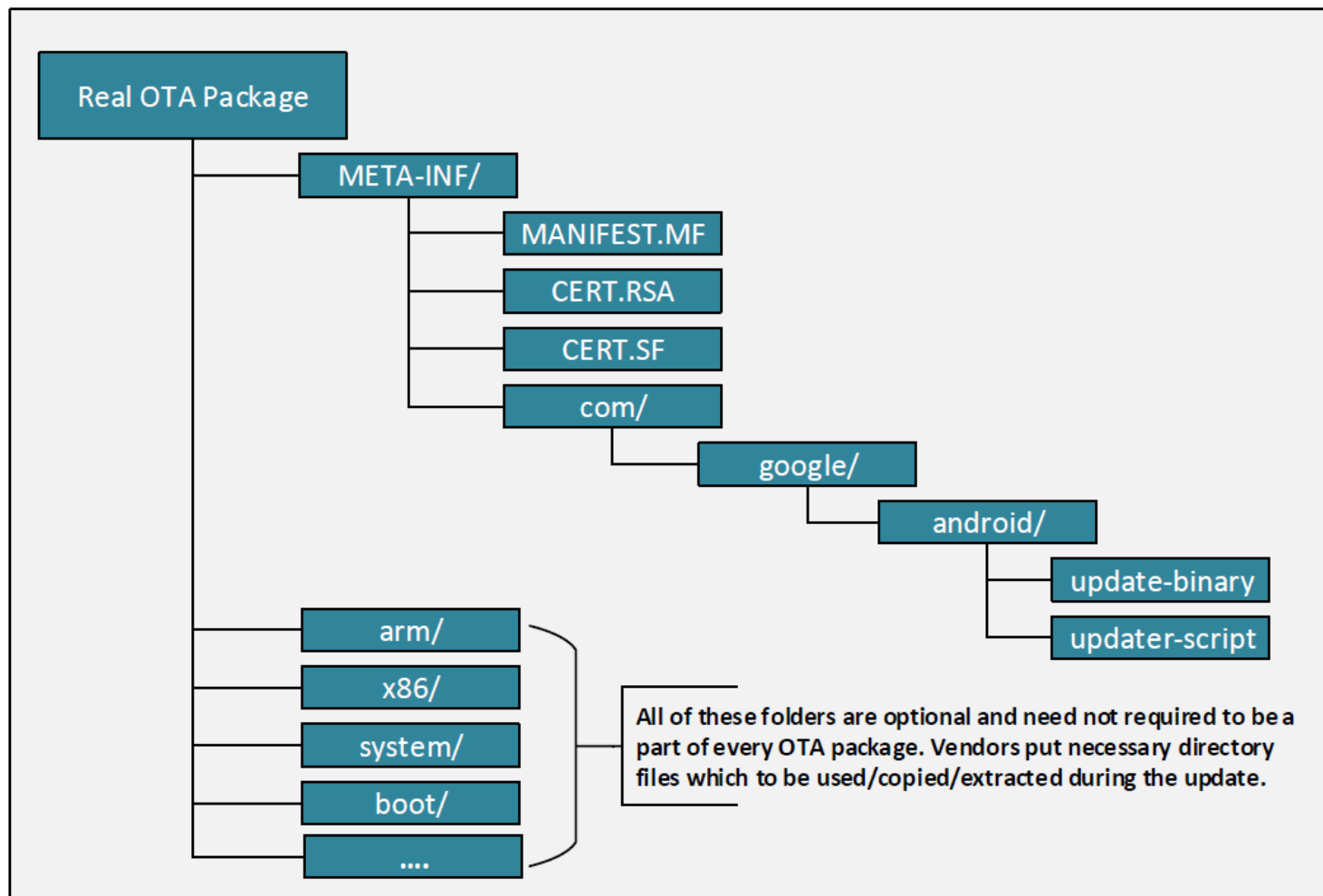
- ❖ **rooting 의 과정속에 Android 시스템 및 운영 체제에 대한 많은 지식이 포함**
  - 시스템 지식을 얻을 수 있는 훌륭한 수단
    - rooting Package 개발 및 Android VM rooting Attack 실습
- ❖ **Android Rooting Attack의 목적**
  - Android OS 내에서 root권한을 얻기

# 배경지식

- ❖ 만약에, Android 장치에 또 다른 OS (rooting Package) 가 설치되어 있다면?
  - 기존의 Android OS가 아닌 다른 OS (rooting Package)로 부팅할 수 있고
  - 다른 OS로 부팅하면 root권한을 얻을 수 있고 모든 파일에 액세스 할 수 있음
- ❖ 사실, Android 장치에 또 다른 OS가 있고 이를 recovery OS라고 함
  - 복구(recovery)의 목적으로 존재하지만 대부분 OS 업데이트에 사용됨
  - 하지만 recovery OS에는 액세스 제어 기능이 있어 사용자가 임의의 명령을 통해 실행하여 업데이트 할 수 없음
  - 대신 외부(Internet 등..)에서 제공하는 Package를 가져올 수 있고 이를 통해 Android OS를 업데이트 할 수 있음
    - Package에는 Android OS 업데이트에 필요한 명령과 파일이 포함
  - 이러한 메커니즘은 OS 업데이트에 흔히 사용되며
    - 이를 OTA(Over-The-Air)업데이트라 하고 이 Package를 OTA Package라 함
- ❖ 본 과제는 recovery OS를 통해 Android Rooting을 하려고 함

# 배경지식

❖ OTA 패키지는 zip 파일이며 그 구조는 아래와 같음



❖ 실습에 주목할 폴더는 META-INF이며 해당 폴더를 작성해야함



# 배경지식

- ❖ META-INF / com / google / android / update-binary :
  - 이 바이너리는 복구 OS에 의해 실행되어 OTA 업데이트를 적용
  - updater-script를 로드하고 실행
- ❖ META-INF / com / google / android / updater-script :
  - 이것은 update-binary에 의해 해석되는 설치 스크립트
  - 업데이트를 적용하기 위해 수행해야하는 조치를 설명

# 실습

## ❖ OTA Package 생성

- 총 3개의 OTA Package 생성
  - META-INF만 포함

## ❖ 파일 구조

- OTA Package
  - META-INF
    - /com
      - /google
        - /android
          - update-binary
          - updater-script

# OTA Package1

## ❖ dummy.sh

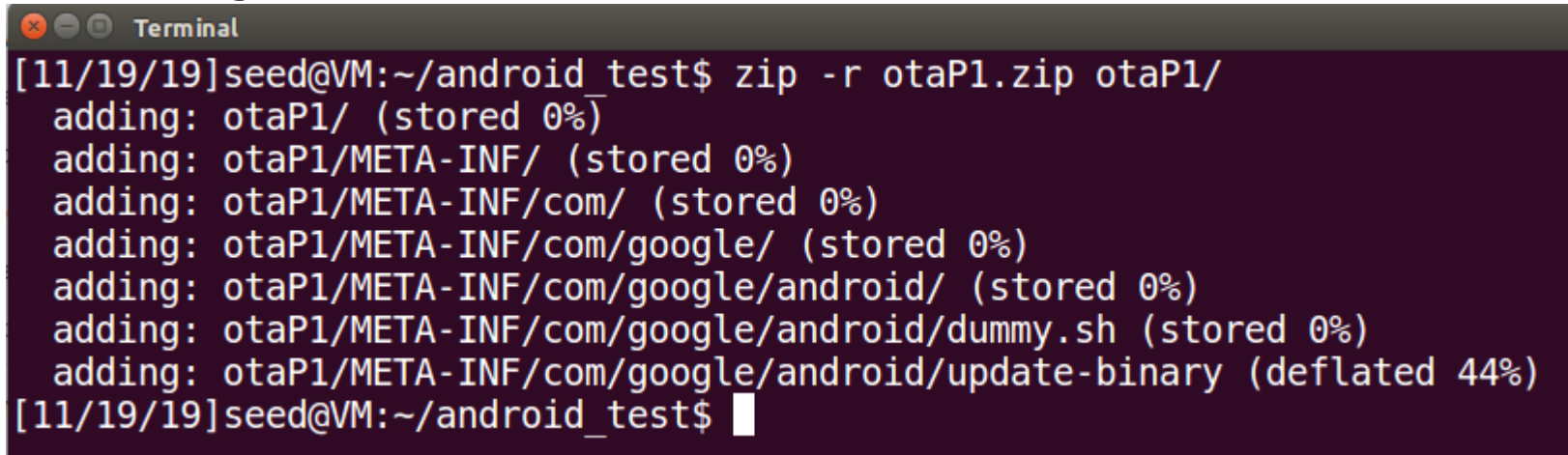
- `echo hello > /system/dummy`

## ❖ update-binary

- 파일이동(copy)
- 실행권한부여
- `sed -i "/return 0/i /system/xbin/dummy.sh" /android/system/etc/init.sh`
  - Explanation:
    - - "-i": edit files in place.
    - - "/return 0/": match the line that has the content return 0.
    - - "i": insert before the matching line.
    - - "/system/xbin/dummy.sh": the content to be inserted. We need to copy the dummy.sh file to the corresponding folder first.
    - - "/android/system/etc/init.sh": the target file modified by "sed".

# OTA Package1

## ❖ OTA Package 생성



```
Terminal
[11/19/19]seed@VM:~/android_test$ zip -r otaP1.zip otaP1/
  adding: otaP1/ (stored 0%)
  adding: otaP1/META-INF/ (stored 0%)
  adding: otaP1/META-INF/com/ (stored 0%)
  adding: otaP1/META-INF/com/google/ (stored 0%)
  adding: otaP1/META-INF/com/google/android/ (stored 0%)
  adding: otaP1/META-INF/com/google/android/dummy.sh (stored 0%)
  adding: otaP1/META-INF/com/google/android/update-binary (deflated 44%)
[11/19/19]seed@VM:~/android_test$
```

## ❖ recovery OS로 전송

- scp명령어 사용
- /tmp로 전송

# OTA Package1

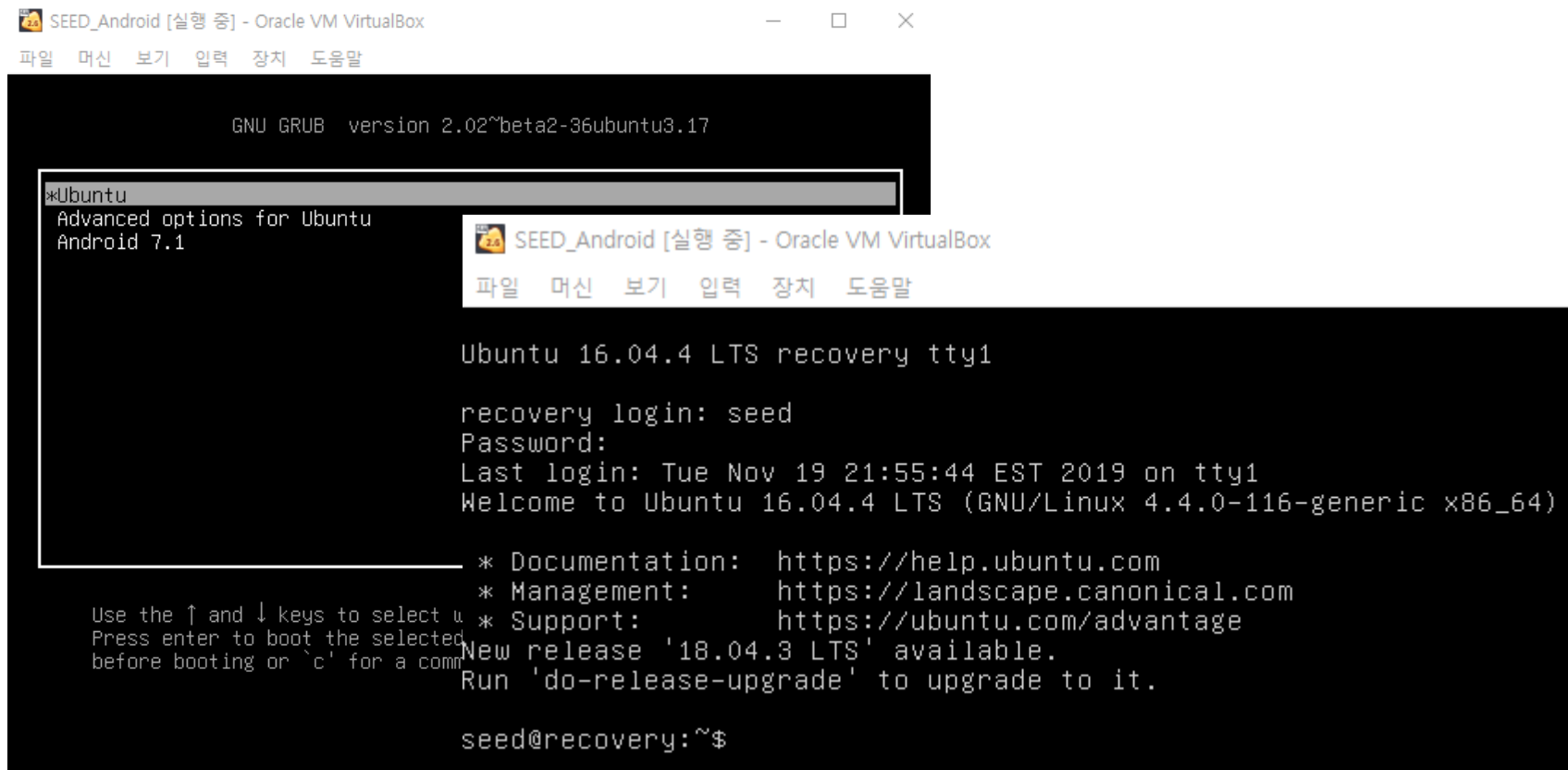
## ❖ scp

- The scp (secure copy) command is the remote version of Unix's cp command. It uses ssh for remote copies.
- Examples:
  - 1. Copy a.txt from SEEDUbuntu16.04\_x32 to recovery OS (located in /home/seed/, aka home directory for seed)
    - `scp ./a.txt seed@<recovery ip>`
  - 2. Copy b.txt from /home/recovery/test/on recovery OS to SEEDUbuntu16.04\_x32
    - `scp seed@<recovery ip>:/home/seed/test/b.txt ./`
    - You will be asked the password (dees) of recovery account when you run these commands.

# OTA Package1

## ❖ recovery OS 접속

- Android\_CSOS 실행후 VirtualBox 로딩화면에서 shift키를 누르고 있다.
- Ubuntu로 접속
- seed로 로그인 (pw : dees)



```
GNU GRUB version 2.02~beta2-36ubuntu3.17
*Ubuntu
Advanced options for Ubuntu
Android 7.1

SEED_Android [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말

Ubuntu 16.04.4 LTS recovery tty1

recovery login: seed
Password:
Last login: Tue Nov 19 21:55:44 EST 2019 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 '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

seed@recovery:~$
```

# OTA Package1

## ❖ recovery OS에서 파일 확인

```
seed@recovery:~$ cd /tmp
seed@recovery:/tmp$ ls -l
total 8
-rw-rw-r-- 1 seed seed 1406 Nov 19 23:17 otaP1.zip
drwx----- 3 root root 4096 Nov 19 23:14 systemd-private-b731909da33d41ea84938ef
1978ede25-systemd-timesyncd.service-j6bfwK
seed@recovery:/tmp$ unzip otaP1.zip
Archive:  otaP1.zip
  creating: otaP1/
  creating: otaP1/META-INF/
  creating: otaP1/META-INF/com/
  creating: otaP1/META-INF/com/google/
  creating: otaP1/META-INF/com/google/android/
  extracting: otaP1/META-INF/com/google/android/dummy.sh
  inflating: otaP1/META-INF/com/google/android/update-binary
seed@recovery:/tmp$ cd otaP1/META-INF/com/google/android/
seed@recovery:/tmp/otaP1/META-INF/com/google/android$ ls -l
total 8
-rw-rw-r-- 1 seed seed  30 Nov 14 10:51 dummy.sh
-rwxrwxr-x 1 seed seed 143 Nov 14 10:53 update-binary
```

## ❖ update-binary 실행

- sudo 명령어로 실행

# OTA Package2

## ❖ my\_app\_process파일에 compile.sh 실행

- /libs/x86에 compiled binary code 생성

## ❖ update-binary

- We need to copy our compiled binary code to the corresponding location inside Android.
- We need to rename the original **app\_process** binary to something else, and then use our code as **app\_process**. The actual name of **app\_process** can be either **app\_process32** or **app\_process64**, depending on the architecture of the device. Our Android VM is a 64-bit device, so the name should be **app\_process64**.



# OTA Package2

## ❖ OTA Package 생성

```
[11/19/19]seed@VM:~/android_test$ zip -r otaP2.zip otaP2/  
adding: otaP2/ (stored 0%)  
adding: otaP2/META-INF/ (stored 0%)  
adding: otaP2/META-INF/com/ (stored 0%)  
adding: otaP2/META-INF/com/google/ (stored 0%)  
adding: otaP2/META-INF/com/google/android/ (stored 0%)  
adding: otaP2/META-INF/com/google/android/update-binary (deflated 58%)  
adding: otaP2/META-INF/com/google/android/my_app_process (deflated 72%)  
[11/19/19]seed@VM:~/android_test$
```

## ❖ recovery OS로 전송

- scp명령어 사용
- /tmp로 전송

## ❖ update-binary 실행

# OTA Package3

## ❖ my\_SU(기존의 SimpleSU)파일에서 compile\_all.sh 실행

- `bash ./compile_all.sh`
- mydaemon파일과 mysu파일의 각각 `/libs/x86`에 **compiled binary code** 생성

## ❖ update-binary

- 파일이동(copy)
- `sed -i` 명령어 사용

# OTA Package3

## ❖ OTA Package 생성

```
[11/19/19]seed@VM:~/android_test$ zip -r otaP3.zip otaP3/  
adding: otaP3/ (stored 0%)  
adding: otaP3/META-INF/ (stored 0%)  
adding: otaP3/META-INF/com/ (stored 0%)  
adding: otaP3/META-INF/com/google/ (stored 0%)  
adding: otaP3/META-INF/com/google/android/ (stored 0%)  
adding: otaP3/META-INF/com/google/android/update-binary (deflated 40%)  
adding: otaP3/META-INF/com/google/android/mydaemon (deflated 60%)  
adding: otaP3/META-INF/com/google/android/mysu (deflated 66%)  
[11/20/19]seed@VM:~/android_test$
```

## ❖ recovery OS로 전송

- scp명령어 사용
- /tmp로 전송

## ❖ update-binary 실행

# OTA Package3

## ❖ 결과

- 다른 방법으로 확인해도 상관없습니다

파일 머신 보기 입력 장치 도움말



Window 1 ▾

```
x86_64:/ # id
uid=0(root) gid=0(root) groups=0(root) context=u:r:su:s0
x86_64:/ #
```

# 평가 기준

## ❖ (1) 수행결과 스크린샷

- OTA Package를 작성하고 파일구조와 내용을 스크린샷으로 첨부
- update-binary 내용을 스크린샷으로 첨부

## ❖ (2) rooting 결과 스크린샷

- Android OS를 성공적으로 rooting 했는지 스크린샷으로 첨부
- Android VM에서 id명령어로 root계정을 탈취했는지 확인

## ❖ (3) Android Rooting Attack 보고서

- 실습내용을 토대로 Android Rooting Attack의 상세한 관찰 결과 작성
- 아래의 내용을 포함하기를 권장함
  - Android 부팅순서 및 응용프로그램
  - rooting
  - recovery OS
  - OTA
- \*\*본 과제의 한계(혹은 제약)

# 제출

## ❖ 11월 20일 (수) ~ 12월 11일 (수)

- 실습 및 과제 내용을 보고서로 제출
- 수업시간에 제출 or 미디어센터 505호로 방문하여 제출
- 부재시 504호 제출
- 표지
  - 과제명, 과목명, 학번/성명, 제출일 반드시 포함

## ❖ 문의

- 이름 : 조재희
- 연락 : [jehee1204@gmail.com](mailto:jehee1204@gmail.com)
  - 메일 제목 앞에 [Rooting]이라고 붙여서 보내주시면 감사하겠습니다

■ 위치 :	505호 출입문	
		조재희

# Thank You !