Android Userland Fuzzing and Exploitation

Lab Setup Guide

Lab Setup Outline

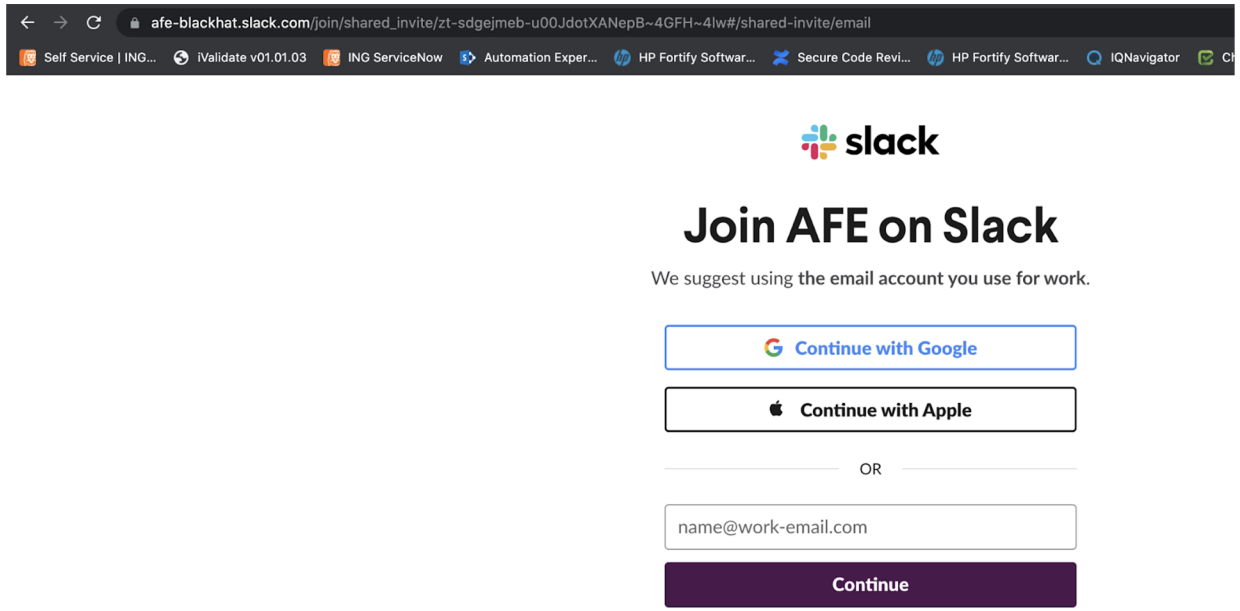
- Lesson 1: Slack Channel Setup
- Lesson 2: Install Vbox and Vbox Addition Package
- Lesson 3: Import Ubuntu/Fuzzing VM
- Lesson 4: Installing Android Studio
- Lesson 5: Installing NDK
- Lesson 6: Import ”Vulnerable Project” in Android Studio
- Lesson 7: Download and run ARM Android VM
- Lesson 8: Connect to phone from Ubuntu VM
- Lesson 9: Corellium Lab Setup

Lesson 1: Slack Channel Setup

- Use the Invite link to join the AFE slack channel
  - https://join.slack.com/t/afe-blackhat/shared_invite/zt-sdgejmeb-u00JdotXANepB~4GFH~4lw

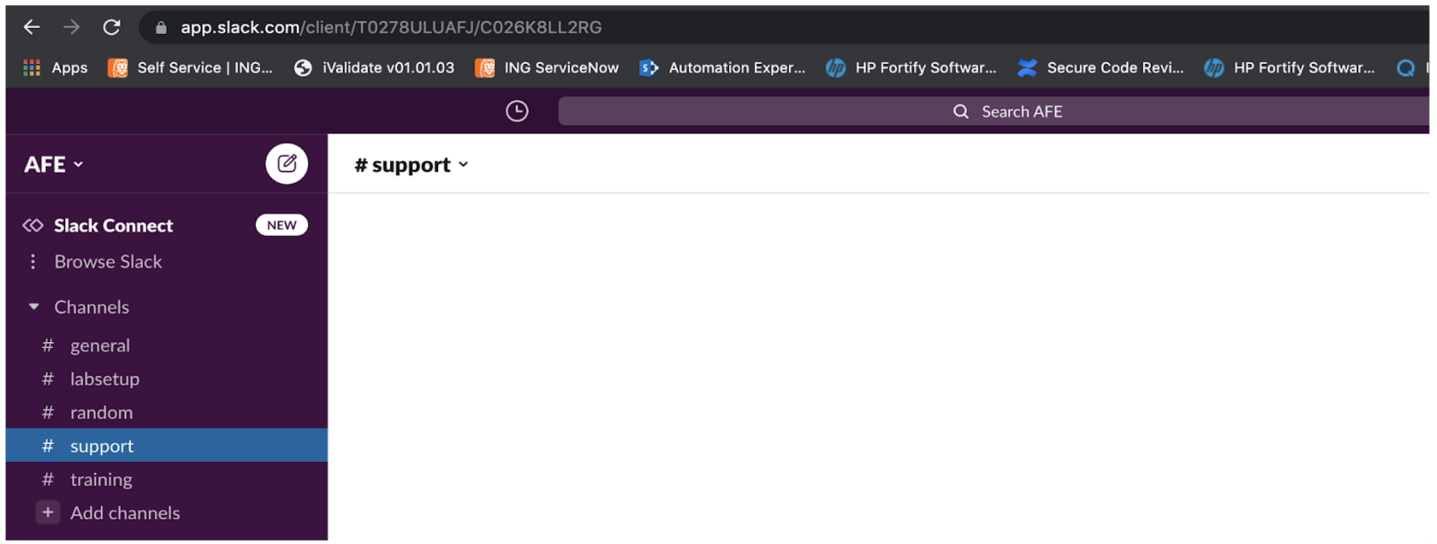
Lesson 1: Slack Channel Setup

- Join the AFE channel either with your Gmail, Apple, or another email account



Lesson 1: Slack Channel Setup

- Once you have joined the slack channel group, you can find multiple channels on the left panel.



- Use separate channels for communication
- If you have any queries on Lab setup, connect us through the Lab setup channel

Lesson 2: Install Vbox and Vbox Addition Package

- Install VirtualBox
- Install VirtualBox Addition Packages

Lesson 2: Install VirtualBox

- Download and Install VirtualBox for your OS
- https://www.virtualbox.org/wiki/Downloads

Lesson 2: Installing VirtualBox Extension Pack

- Download and install the VirtualBox Extentsion Pack
- https://www.virtualbox.org/wiki/Downloads
- Download the ”All support platform” file
- Double click this file to import
- NOTE: The version number is very important, use the newest version

Lesson 2: Wrap-up

- Installed VirtualBox
- Installed VirtualBox Extension Pack

Lesson 3: Import Ubuntu/Fuzzing VM

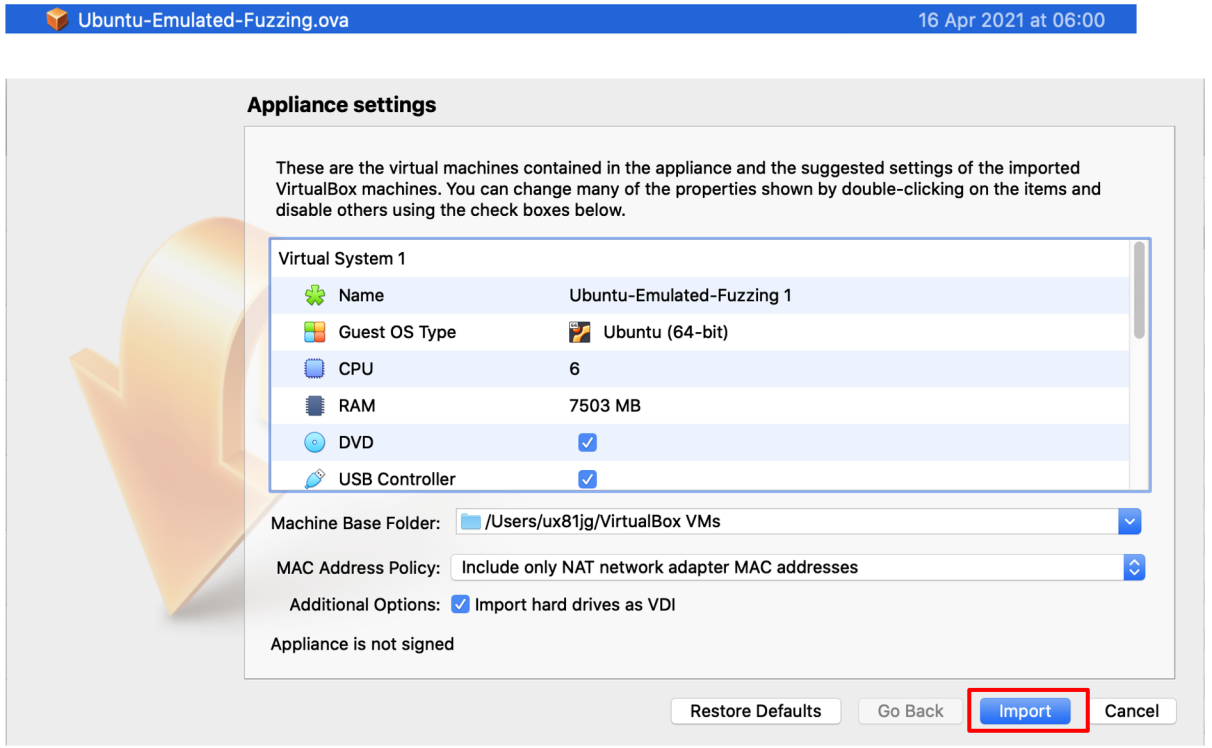
- Import Ubuntu/Fuzzing VM

Lesson 3: Import Ubuntu/Fuzzing VM

- Download the training VM
- https://drive.google.com/file/d/1AqBUVohKMxI22fMnNe4l1Fq7j4DQLzb7/view?usp=sharing

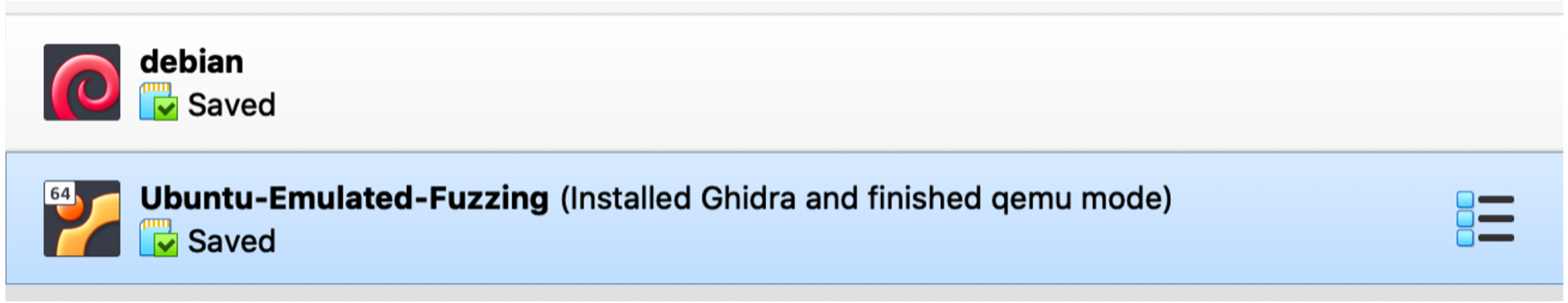
- Import the file into VirtualBox by double clicking on the file

- Press the “Import” button to import



Lesson 3: Import Ubuntu/Fuzzing VM

- Downloaded the VM image
- Imported the VM image into VirtualBox

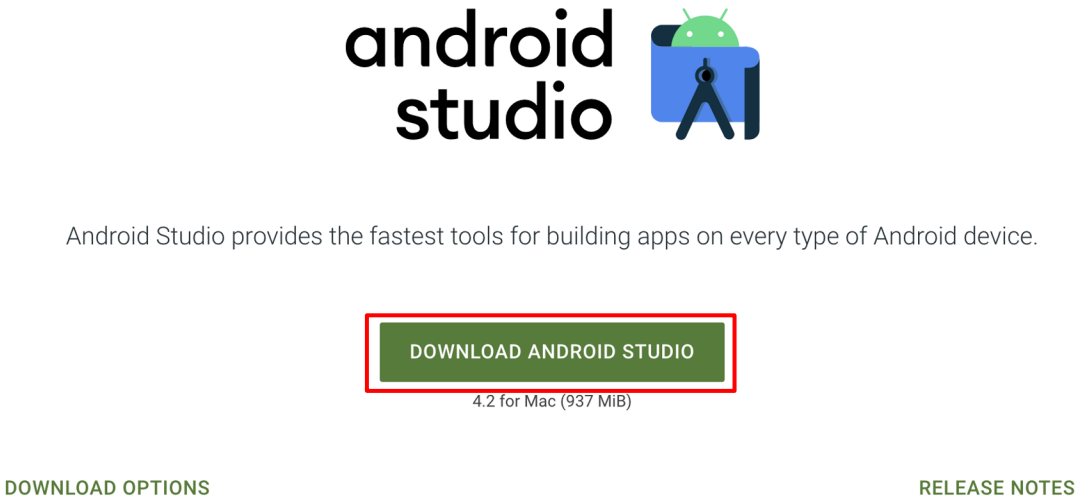


Lesson 4: Installing Android Studio

- Download Android Studio
- Install Android Studio

Lesson 4: Installing Android Studio

- Download Android Studio for you Operating System
- https://developer.android.com/studio
- Install according your Operating System



Lesson 4: Wrap-up

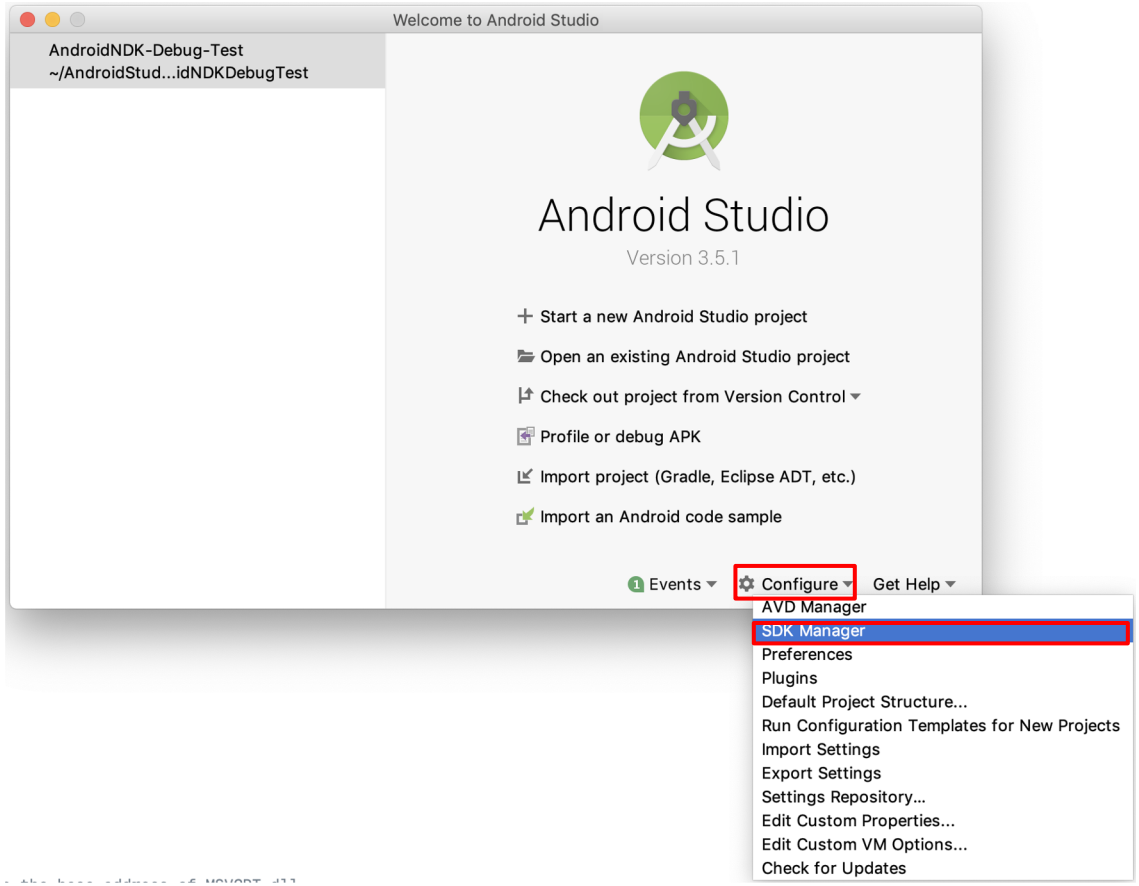
- Download Android Studio for your Operating System
- Install Android Studio

Lesson 5: Installing NDK

- Downloading and installing NDK

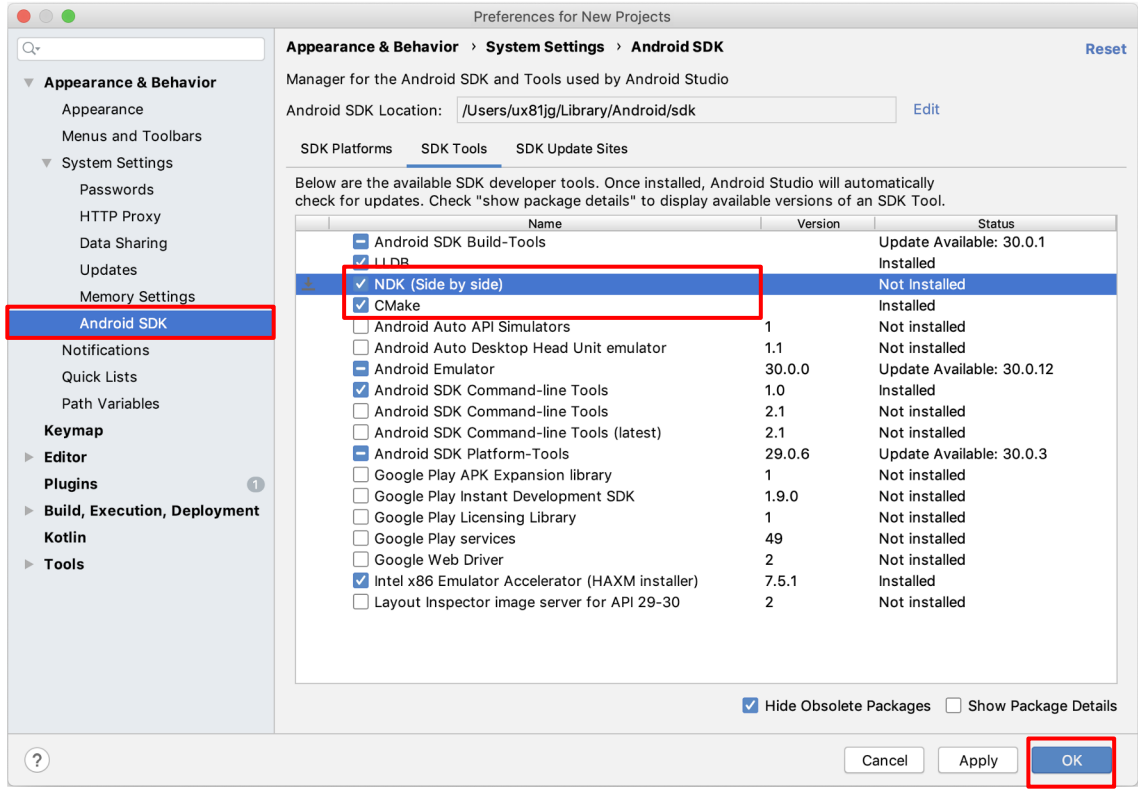
Lesson 5: Installing NDK

- Open Android Studio
- Click the “Configure” option
- Select SDK Manager from the options



Lesson 5: Installing NDK

- Click on “SDK Tool” tab
- Check the “NDK” check box
- Check the “CMake” checkbox
- Press the ”Ok” button to install



Lesson 5: Wrap-up

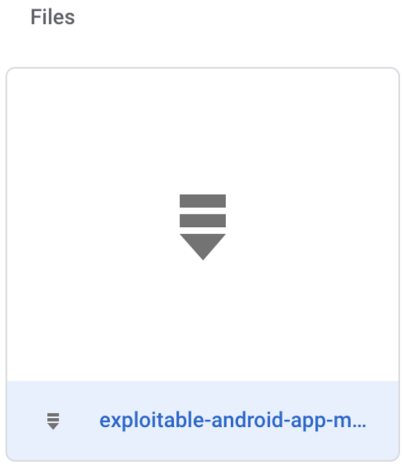
- Download Android Studio for your Operating System
- Install Android Studio

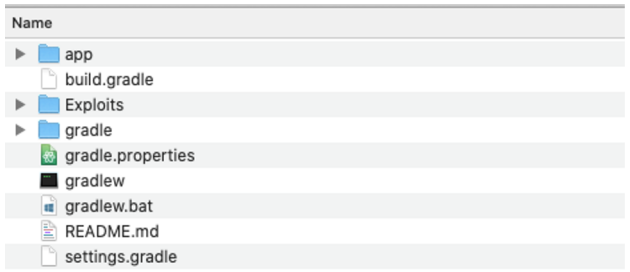
Lesson 6: Import ”Vulnerable Project” in Android Studio

- Download ”vulnerable project”
- https://drive.google.com/file/d/1V7-JAMxH2u5c6nfHxtcYfgcCuI2jCsJK/view?usp=sharing
- Import project

Lesson 6: Download ”Vulnerable Project”

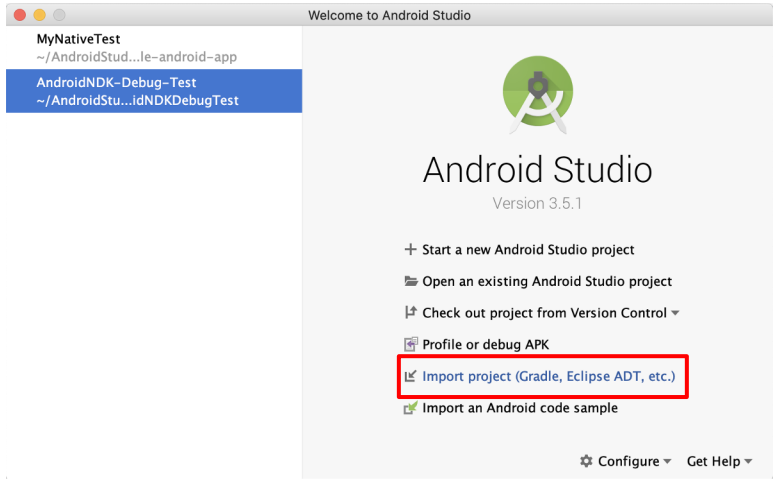
- Download the “vulnerable project”
- https://drive.google.com/file/d/1V7-JAMxH2u5c6nfHxtcYfgcCuI2jCsJK/view?usp=sharing
- Unzip the project

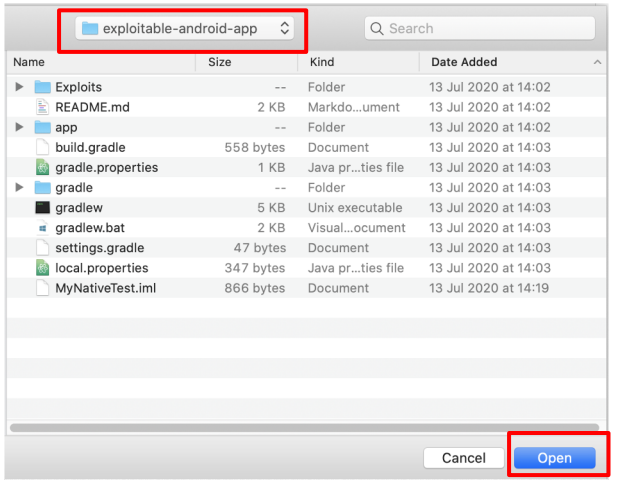




Lesson 6: Import ”Vulnerable Project” in Android Studio

- Open Android Studio
- Click “Import Project”





Lesson 6: Wrap-up

- Download Android Studio for your Operating System
- Install Android Studio

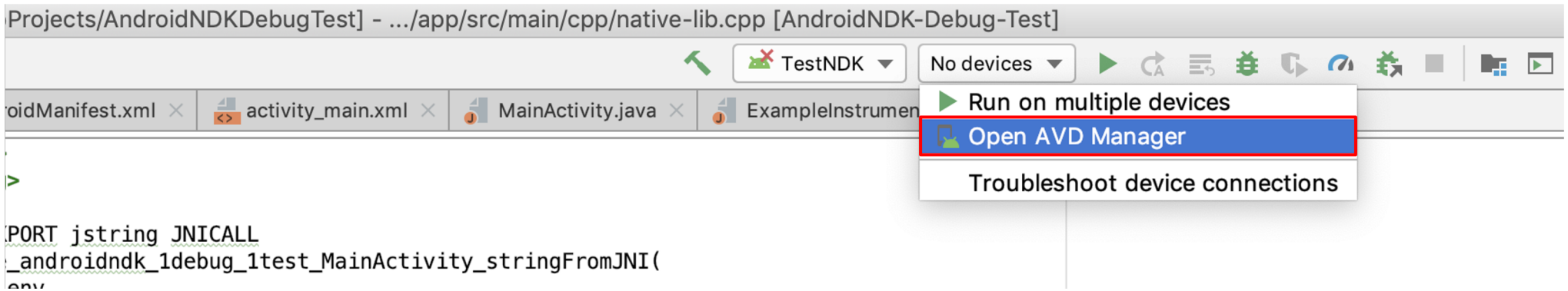
Lesson 7: Download and run ARM Android VM

- Download ARM Android VM
- Running the VM

Lesson 7: Download and run ARM Android VM

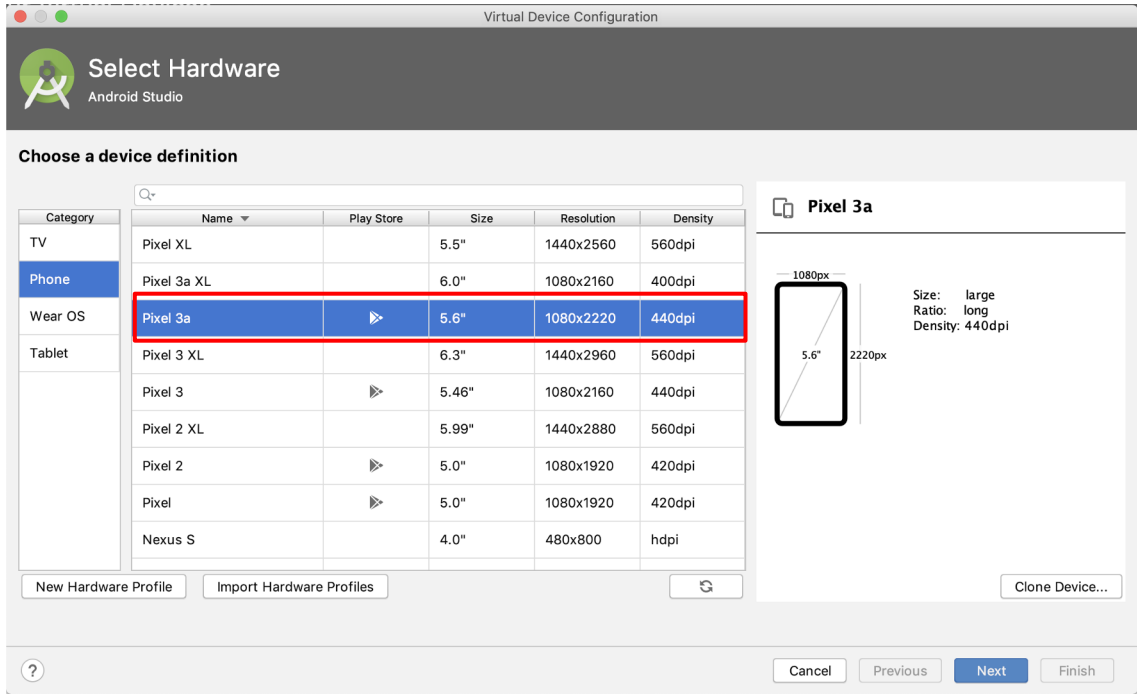
- Open Android Studio
- Select the newly imported project and open

- On the top choose the “No devices” and select “Open AVD Manager”



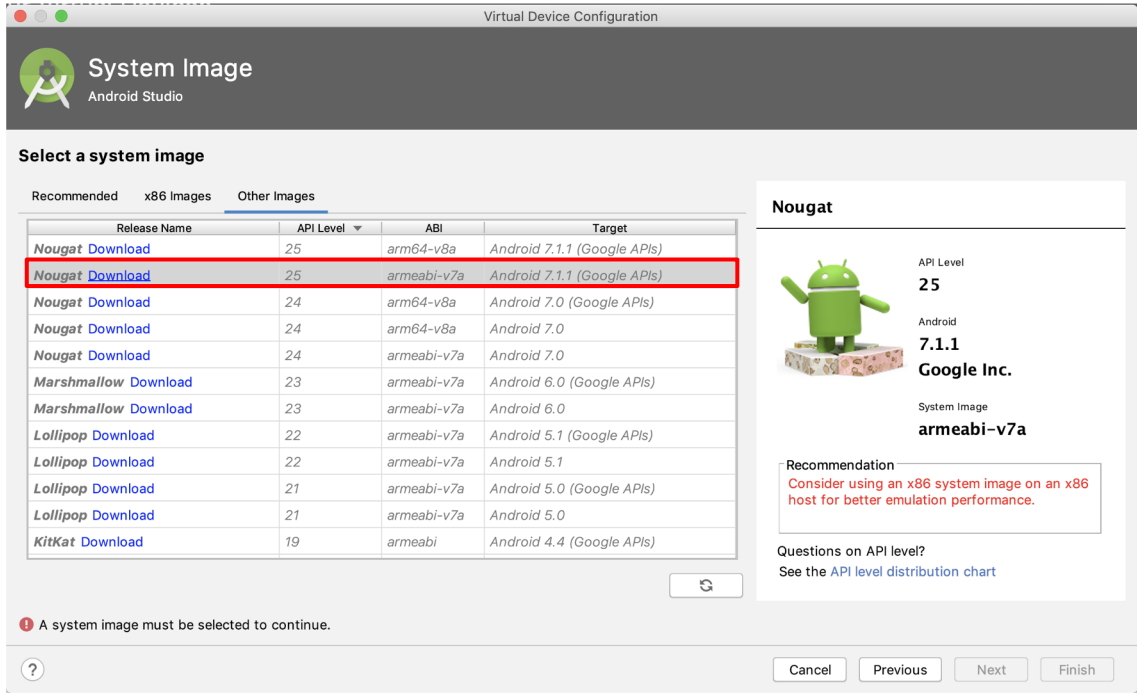
Lesson 7: Download and run ARM Android VM

- Select the Hardware we would like to run
- Choose Pixel 3a as shown in picture
- Click next



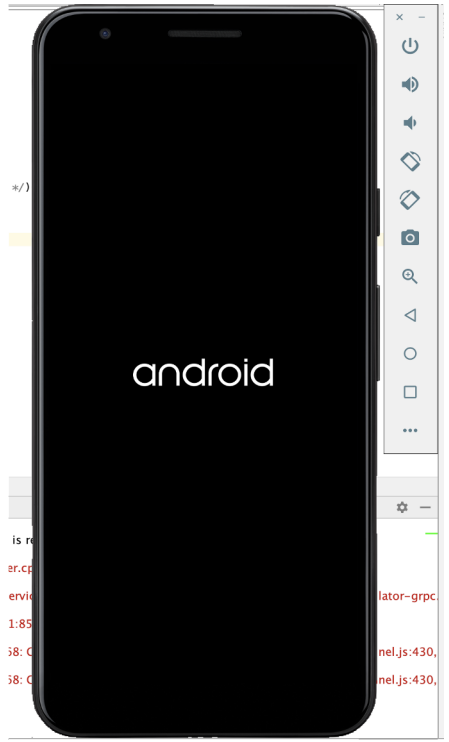
Lesson 7: Download and run ARM Android VM

- Select the correct image
  - **Nougat**
  - **API Level 25**
  - **Armabi-v7a**
  - **Android 7.1.1**
- Click “Download”
- Click next after download



Lesson 7: Download and run ARM Android VM

- Running the VM is easy, click the play button in the ribbon
- This should launch the VM



NOTE: It will take some time for the VM to boot up and install the app

Lesson 7: Wrap-up

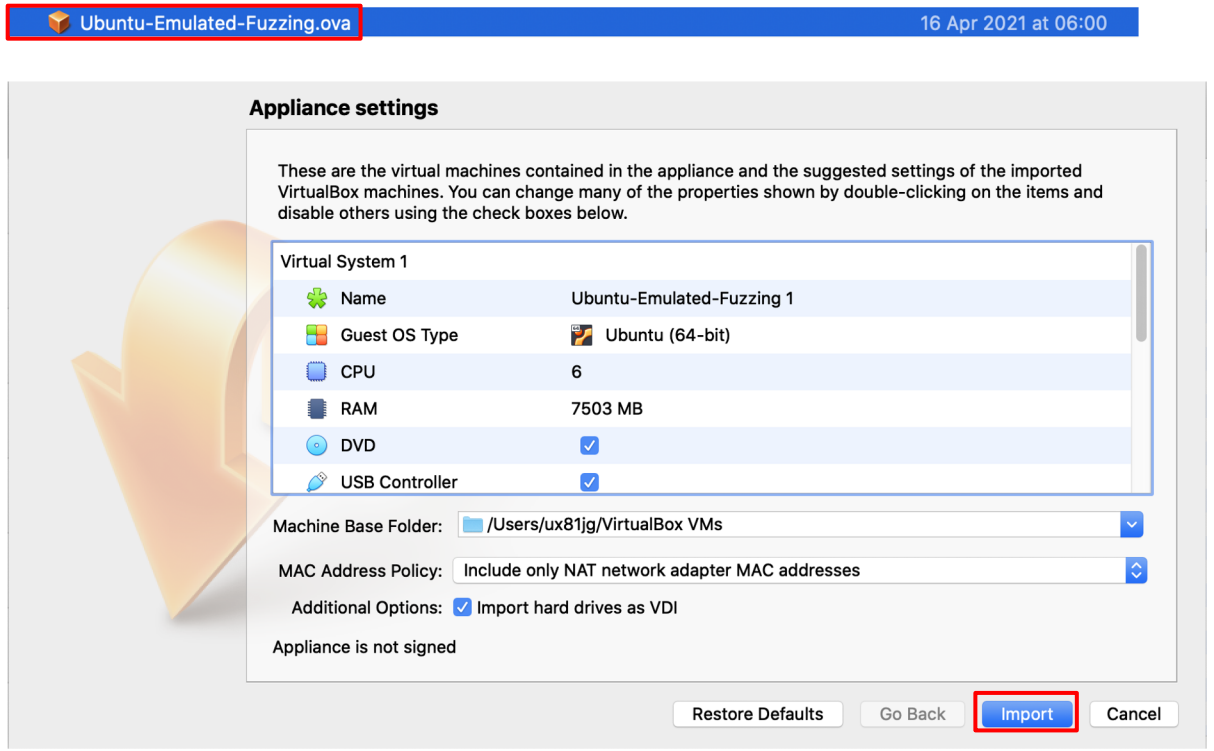
- Download Android Studio for your Operating System
- Install Android Studio

Lesson 8: Connect to phone from Ubuntu VM

- Download the training VM
- https://drive.google.com/file/d/1AqBUVohKMxI22fMnNe4l1Fq7j4DQLzb7/view?usp=sharing

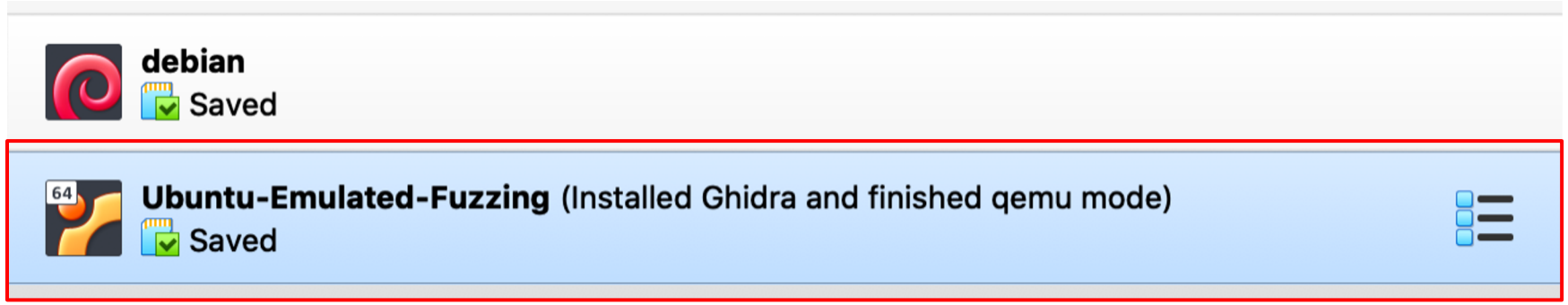
- Import the file into VirtualBox by double clicking on the file

- Press the “Import” button to import



Lesson 8: Connect to phone from Ubuntu VM

- Open VirtualBox
- Double click on the imported machines to run it

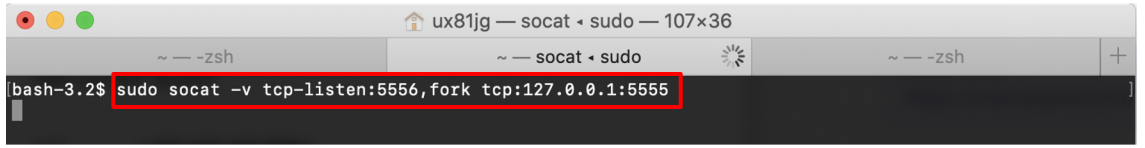


Lesson 8: Connect to phone from Ubuntu VM

- ADB connection forward
- Android App VM Connection forward
- Android Device GDB Server forward

Lesson 8: ADB connection forward

- Download SOCAT
  - for Linux(Ubuntu)
    - apt install socat
  - for Mac:
    - brew install socat
  - For Windows
    - see next slides

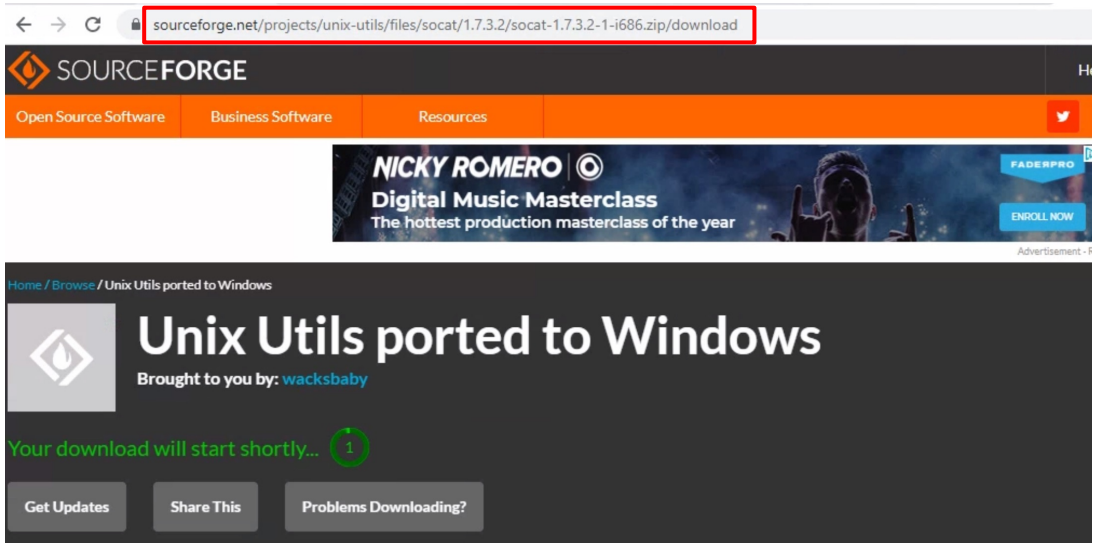


- Open terminal and type:
- sudo socat -v tcp-listen:5556,fork tcp:127.0.0.1:5555

- This will forward all ADB connection on localhost to all interfaces on port 5556

Lesson 8: ADB connection forward for Windows

- Download the SOCAT using below link
  - https://sourceforge.net/projects/unix-utils/files/socat/1.7.3.2/socat-1.7.3.2-1-i686.zip/download
- Extract the downloaded zip file

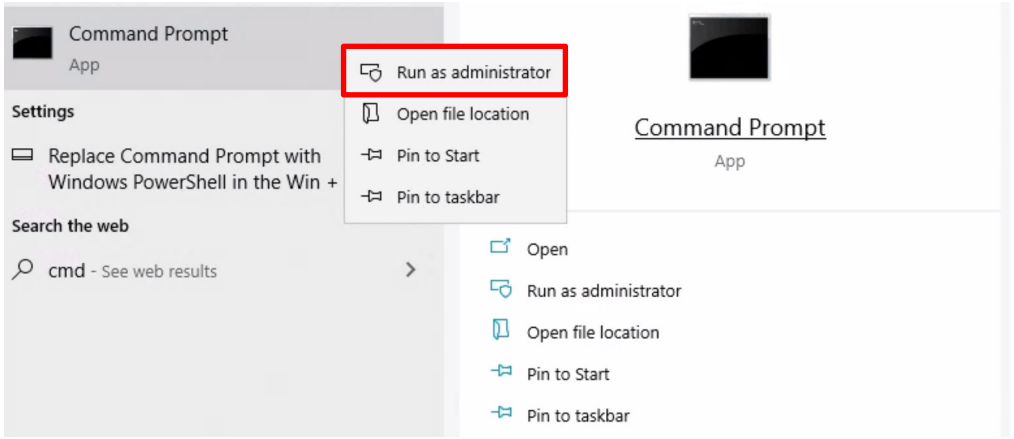


NOTE: This step is only required if you use Windows as your base machine!

Lesson 8: ADB connection forward for Windows

- Open command prompt run as administrator
- Go to unzipped location and type:
- socat.exe -v tcp-listen:5556,fork tcp:127.0.0.1:5555

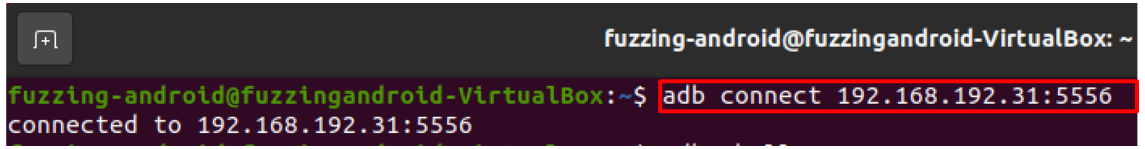
- This will forward all ADB connection on localhost to all interfaces on port 5556



NOTE: This step is only required if you use Windows as your base machine!

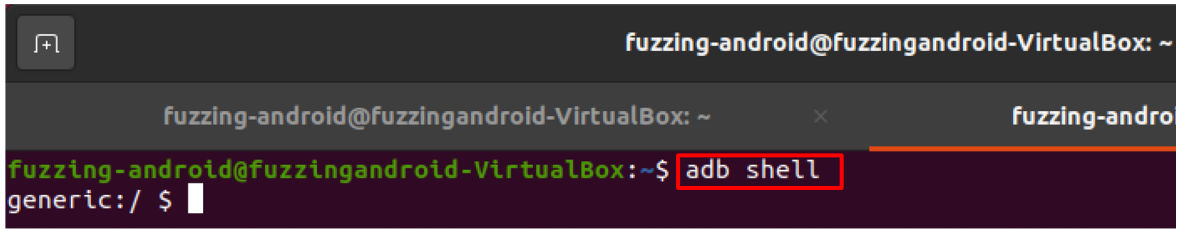
Lesson 8: ADB connection forward

- Open you VirtualBox VM
- In the Ubuntu VM open a terminal
- Connect to you host adb from the VM by typing the following command



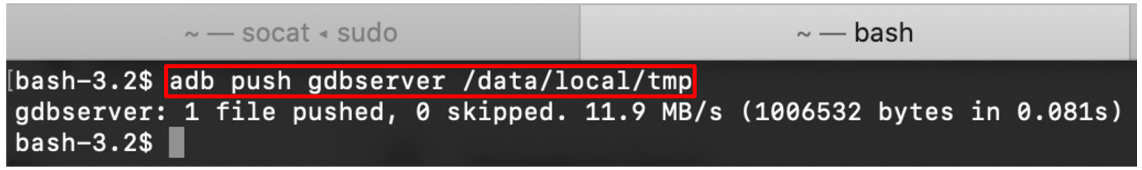
- “**adb connect 192.168.192.31:5556**”
- **NOTE: Your ip address is your HOST machine IP**

- Now type “adb shell” and you should be dropped into a shell of the Android VM



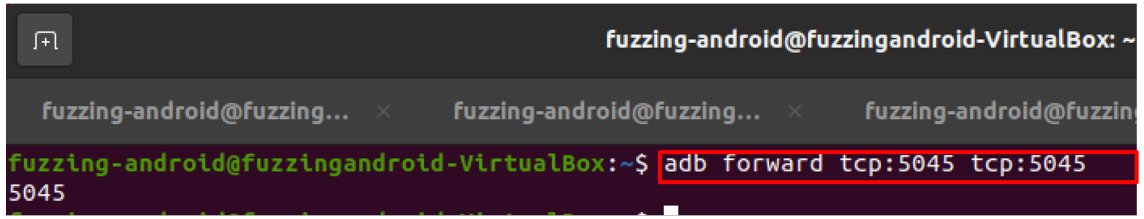
Lesson 8: Download and Upload GDBServer to device

- Download the precompiled version of GDBServer
- https://drive.google.com/file/d/19QLHCRnMh16AnJcmM_iYQW87xIgh9bPa/view?usp=sharing
- Push the file to the device with the following command
- “adb push gdbserver /data/local/tmp”



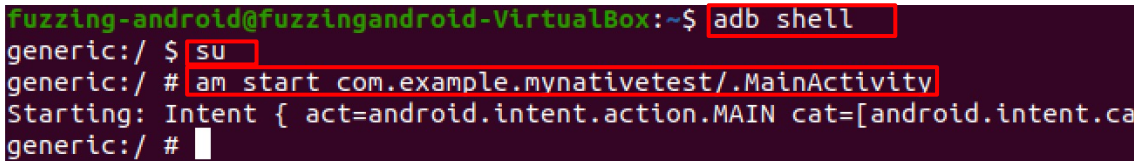
Lesson 8: Download and Upload GDBServer to device

- Form the connect adb VM now lets forward the gdbserver port to our Ubuntu VM
- Type the following command
- “Adb forward tcp:5045 tcp:5045”



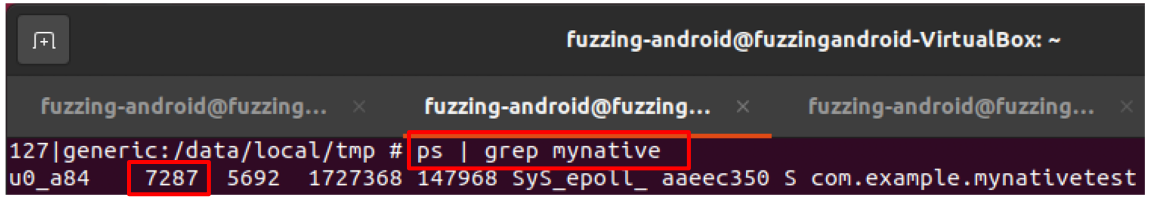
Lesson 8: Download and Upload GDBServer to device

- First start the app then “adb shell” into the Android VM Device
- Type “su” to become root
- Start our app with the following command
- “am start com.example.mynativetest/.MainActivity”

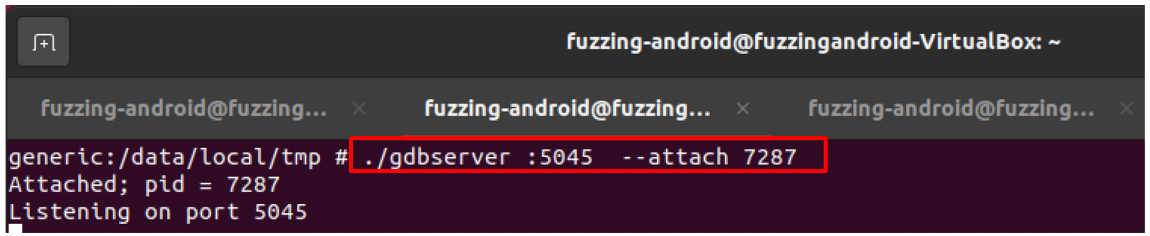


- Go to /data/local/tmp
- There we have our gdbserver
- Make it executable with chmod
- “chmod +x gdbserver”

- Type “ps | grep mynative” to find your process pid



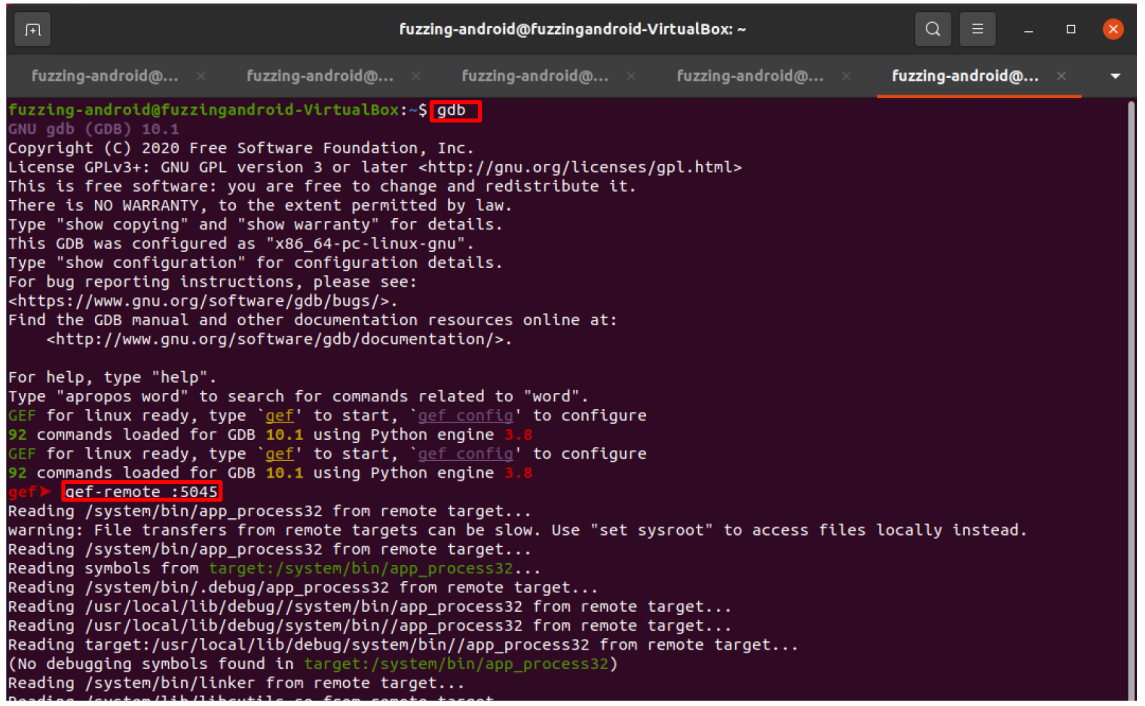
- Then run gdbserver and attach to our PID
- ./gdbserver. :5045 --attach PID



Lesson 8: Download and Upload GDBServer to device

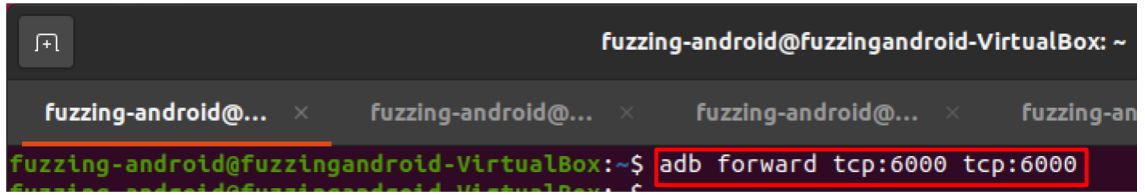
- Now open another terminal and run gdb
- In gdb run “gef-remote :5045”

- You should see the same as the image on the right

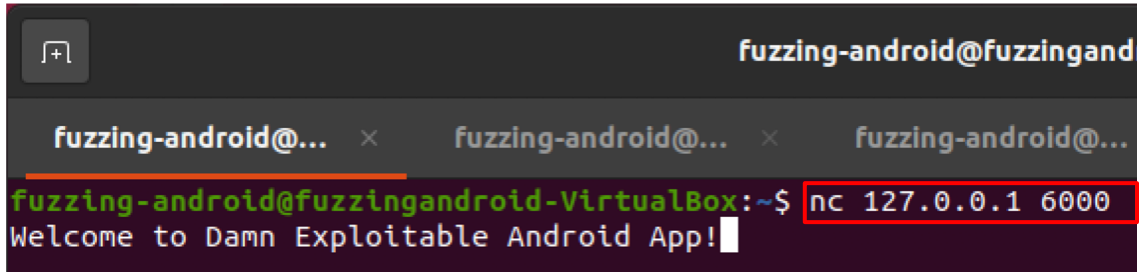


Lesson 8: Forward Vulnerable App port to Ubuntu VM

- Forward app listening port to Ubuntu VM local loopback
- Now open another terminal and type
  - “adb forward tcp:6000 tcp:6000”



- Now connect from the Ubuntu VM to test
- “nc 127.0.0.1 6000”



Lesson 8: Wrap Up

- Forwarded adb from host to Ubuntu VM
- Downloaded precompiled GDBServer
- Forward GDBServer port to Ubuntu VM
- Pushed GDBServer to Android VM
- Attached GDBServer to the running Application
- Forward Vulnerable App port to Ubuntu VM
- Test connection to vulnerable application

Lesson 9: Corellium Lab Setup

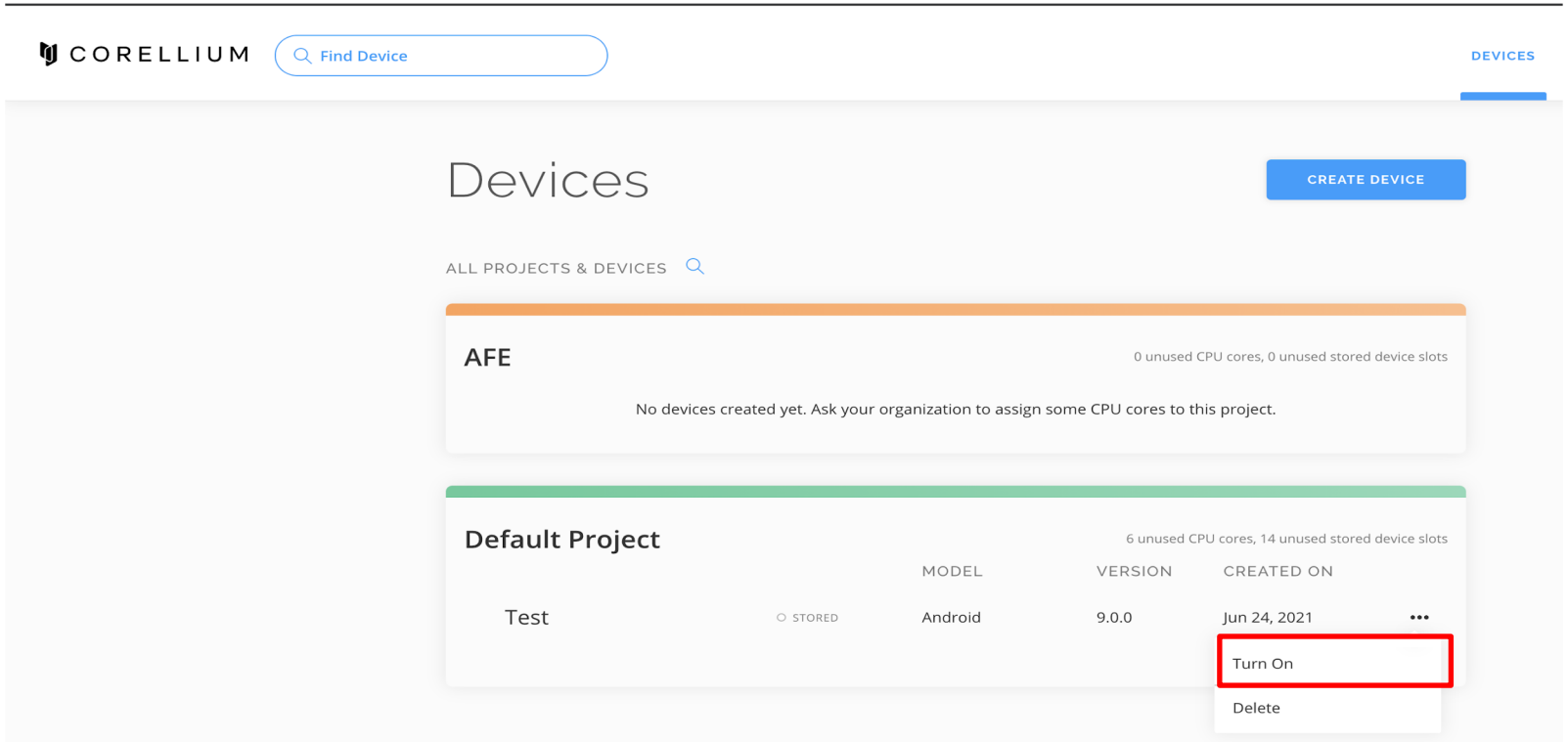
- Login into Corellium portals using the below link:

https://afe.enterprise.corellium.com/login

- You will be provided access through our Slack Group

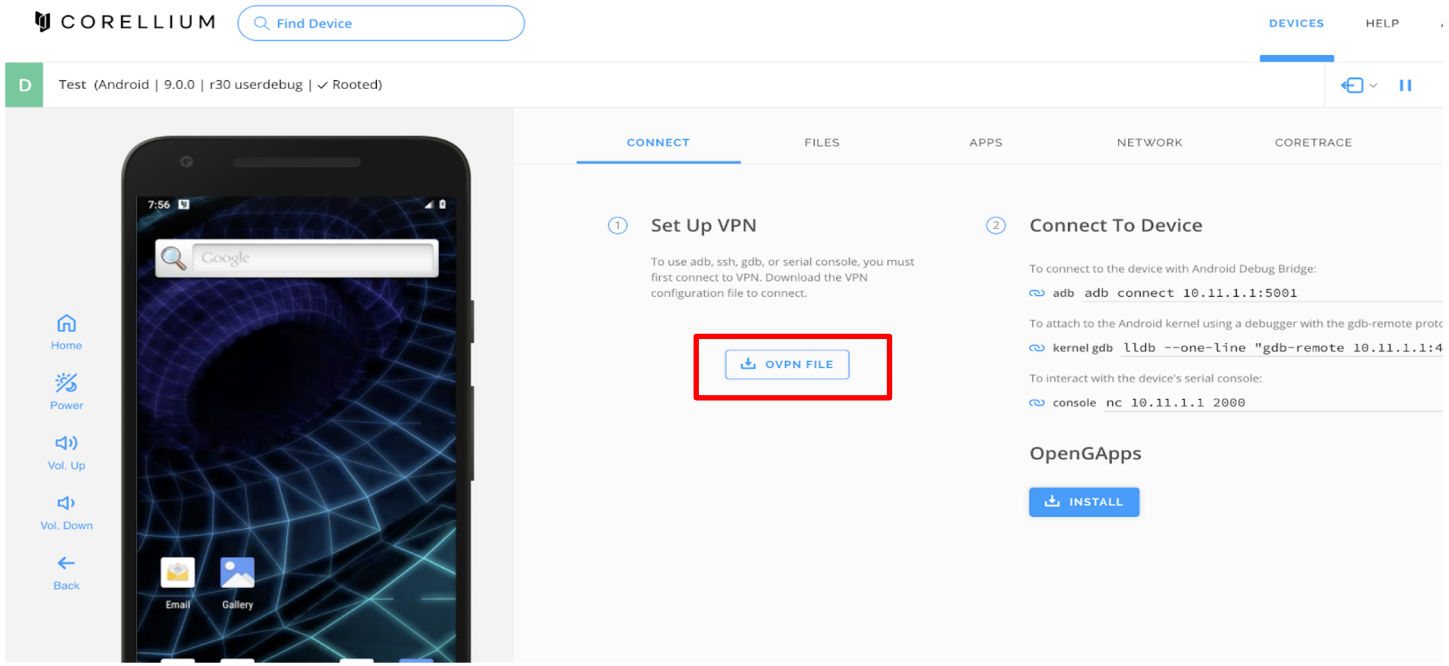
Lesson 9: Corellium Lab Setup

- Once you logged into the Corellium portal, turn on the device



Lesson 9: Corellium Lab Setup

- Download ovpn file from your fuzzing VM




```
fuzzing-android@fuzzingandroid-VirtualBox:~/Downloads$ ls
corellium.ovpn
```

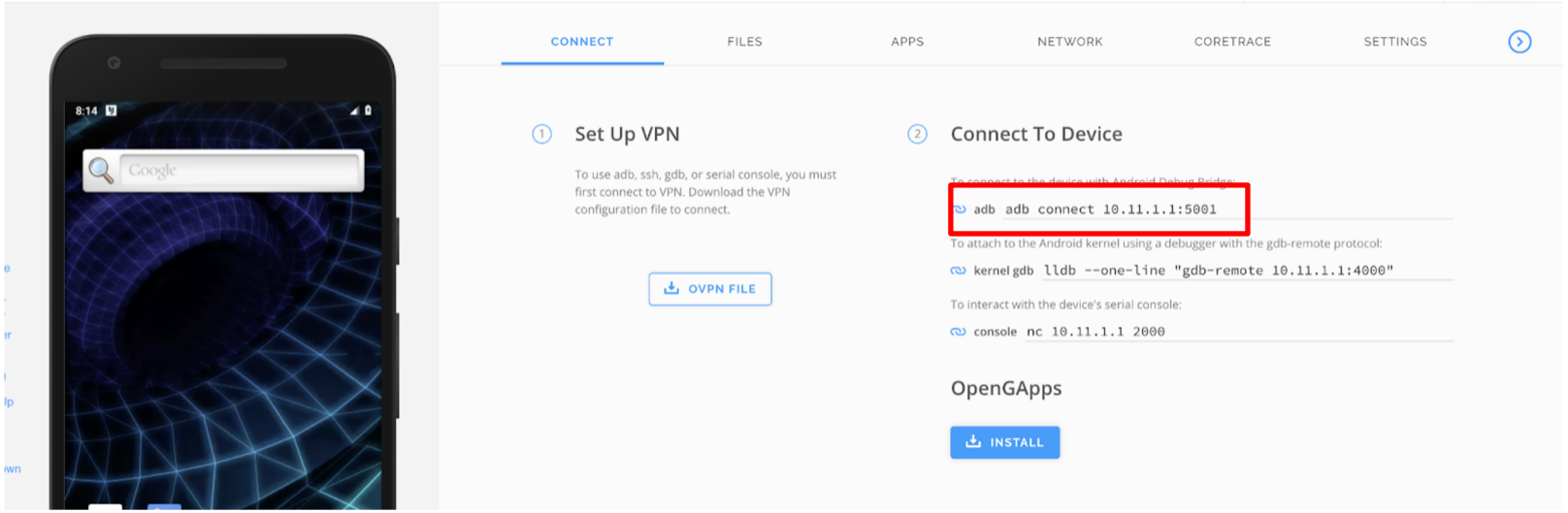
Lesson 9: Corellium Lab Setup

- Use openvpn to connect to the Corellium lab device
  - sudo openvpn --config corellium.ovpn

```
fuzzing-android@fuzzingandroid-VirtualBox:~/Downloads$ sudo openvpn --config corellium.ovpn
[sudo] password for fuzzing-android:
Sat Jul  3 13:43:28 2021 OpenVPN 2.4.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Apr 27 2021
Sat Jul  3 13:43:28 2021 library versions: OpenSSL 1.1.1f  31 Mar 2020, LZO 2.10
Sat Jul  3 13:43:28 2021 TCP/UDP: Preserving recently used remote address: [AF_INET]18.218.250.49:16021
Sat Jul  3 13:43:28 2021 UDP link local: (not bound)
Sat Jul  3 13:43:28 2021 UDP link remote: [AF_INET]18.218.250.49:16021
Sat Jul  3 13:43:29 2021 [openvpn-ee351377-19a5-44af-b01f-cbcfa04cca7f] Peer Connection Initiated with [AF_INET]18.218.250.49:16021
Sat Jul  3 13:43:30 2021 TUN/TAP device tap0 opened
Sat Jul  3 13:43:30 2021 /sbin/ip link set dev tap0 up mtu 1500
Sat Jul  3 13:43:30 2021 /sbin/ip addr add dev tap0 10.11.3.3/22 broadcast 10.11.3.255
Sat Jul  3 13:43:30 2021 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Sat Jul  3 13:43:30 2021 Initialization Sequence Completed
```

Lesson 9: Corellium Lab Setup

- To connect with the device, copy the command from the Corellium interface




```
fuzzing-android@fuzzingandroid-VirtualBox:~$ adb connect 10.11.1.1:5001
connected to 10.11.1.1:5001
fuzzing-android@fuzzingandroid-VirtualBox:~$ adb shell
shell:/$
```

Summary of Lab Setup

- Joined the training Slack Group
- Installed Virtual Box and Guest Additions
- Imported Ubuntu/Fuzzing VM
- Installed Android Studio
- Installed NDK
- Imported ”Vulnerable Project” in Android Studio
- Downloaded and tested ARM Android VM
- Connected the phone from Ubuntu VM
- Lesson 9: Corellium Lab Setup

Summary of Lab Setup

- For any other questions please join the Slack group!