CuckooDroid

Installation and requirements

test configuration: Ubuntu 18.04 e Android 4.1 for the guest machine.

It is strongly suggested to use a python virtualenv:

```
mkdir -p py_venvs
cd py_venvs/
python2 -m virtualenv cuckoodroid_avd_env
cd cuckoodroid_avd_env/
source bin/activate
```

Download

```
$ git config --global user.email "you@example.com"

$ git config --global user.name "Your Name"

$ git clone --depth=1 https://github.com/cuckoobox/cuckoo.git cuckoo -b 1.2

$ cd cuckoo

$ git remote add droid https://github.com/idanr1986/cuckoo-droid

$ git pull --allow-unrelated-histories --no-edit -s recursive -X theirs droid master

$ cat conf-extra/processing.conf >> conf/processing.conf

$ cat conf-extra/reporting.conf >> conf/reporting.conf

$ rm -r conf-extra

$ echo "protobuf" >> requirements.txt
```

Configuration files edit

conf/cuckoo.conf

```
# Specify the name of the machinery module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machinery = avd

[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery configuration.
ip = 127.0.0.1
```

conf/avd.conf

```
[avd]
```

```
#Path to the local installation of the android
emulator

emulator_path = <add> ( /home/USER/Android/Sdk/emulator/emulator )


#Path to the local installation of the adb - android debug bridge
utility.

adb_path = <add> ( /home/USER/Android/Sdk/platform-tools/adb )


#Path to the emulator machine files is
located

avd_path = <add home_path>/.android/avd ( /home/USER/.android/avd )


#name of the reference machine that is used to
duplicate

reference_machine = aosx
# Specify a comma-separated list of available machines to be used. For
each

# specified ID you have to define a dedicated section containing the
details

# on the respective machine. (E.g.
aosx_1,aosx_2,aosx_3)

#currently supports only 1 machine for network
limitations

machines =aosx_1


[aosx_1]

# Specify the label name of the current machine as specified in
your

# aosx_1 configuration.
```

```
label = aosx_1


# Specify the operating system platform used by current
machine

platform = android


# Specify the IP address of the current virtual machine. Make sure
that the

# IP address is valid and that the host machine is able to reach it. If
not,

# the analysis will fail.

# its always 127.0.0.1 because android emulator networking configurations this the loopback
of the

host machine

ip = 127.0.0.1


#Specify the port for the emulator as your adb sees
it.

emulator_port=5554


#10.0.2.2 is the loopback of the host machine very
important!!!

resultserver_ip = 10.0.2.2


resultserver_port = 2042
```

conf/auxiliary.conf

```
[sniffer]

# Enable or disable the use of an external sniffer (tcpdump)
[yes/no].

enabled = yes
```

conf/processing.conf

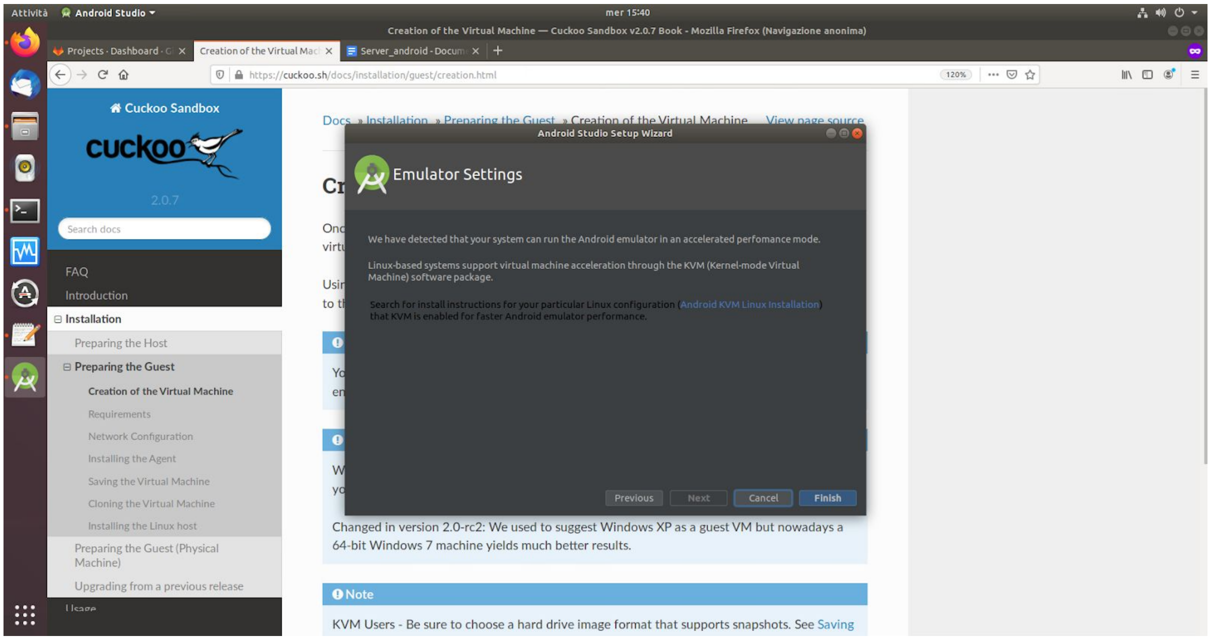
```
[droidmon]

enabled = yes


[googleplay]

enabled = no

android_id = <add android_id>

google_login = <add google_login>

google_password = <add
google_password>


[apkinfo]

enabled = yes

#Decompiling dex with androguard in a heavy operation and for a big
dex's

#he can really consume performance from the cuckoo host ,so it's recommended to limit the
size of

dex that you will decompile

#decompilation_threshold=2000000
```

conf/reporting.conf

```
[reporthtml]

enabled = no

[reportandroidhtml]

enabled = yes
```

Requirements

#installing android studio to use AVD
sudo snap install android-studio --classic #at the time of this writing version is 3.5.2.0
#check if KVM is enabled(see https://developer.android.com/studio/run/emulator-acceleration?utm_source=android-studio#vm-linux or the version at the time of this writing https://web.archive.org/web/20191205093426/https://developer.android.com/studio/run/emulator-acceleration?utm_source=android-studio)



Attività
Android Studio
mer 15:40
Creation of the Virtual Machine — Cuckoo Sandbox v2.0.7 Book - Mozilla Firefox (Navigazione anonima)
Projects · Dashboard
Creation of the Virtual Mac
Server_android - Docum
https://cuckoo.sh/docs/installation/guest/creation.html
120%
Cuckoo Sandbox
cuckoo
2.0.7
Search docs
FAQ
Introduction
Installation
Preparing the Host
Preparing the Guest
Creation of the Virtual Machine
Requirements
Network Configuration
Installing the Agent
Saving the Virtual Machine
Cloning the Virtual Machine
Installing the Linux host
Preparing the Guest (Physical Machine)
Upgrading from a previous release
Usage
Docs » Installation » Preparing the Guest » Creation of the Virtual Machine
View page source
Android Studio Setup Wizard
Emulator Settings
Previous
Next
Cancel
Finish
Note

We have detected that your system can run the Android emulator in an accelerated perfomance mode.

Linux-based systems support virtual machine acceleration through the KVM (Kernel-mode Virtual Machine) software package.

Search for install instructions for your particular Linux configuration (Android KVM Linux Installation) that KVM is enabled for faster Android emulator performance.

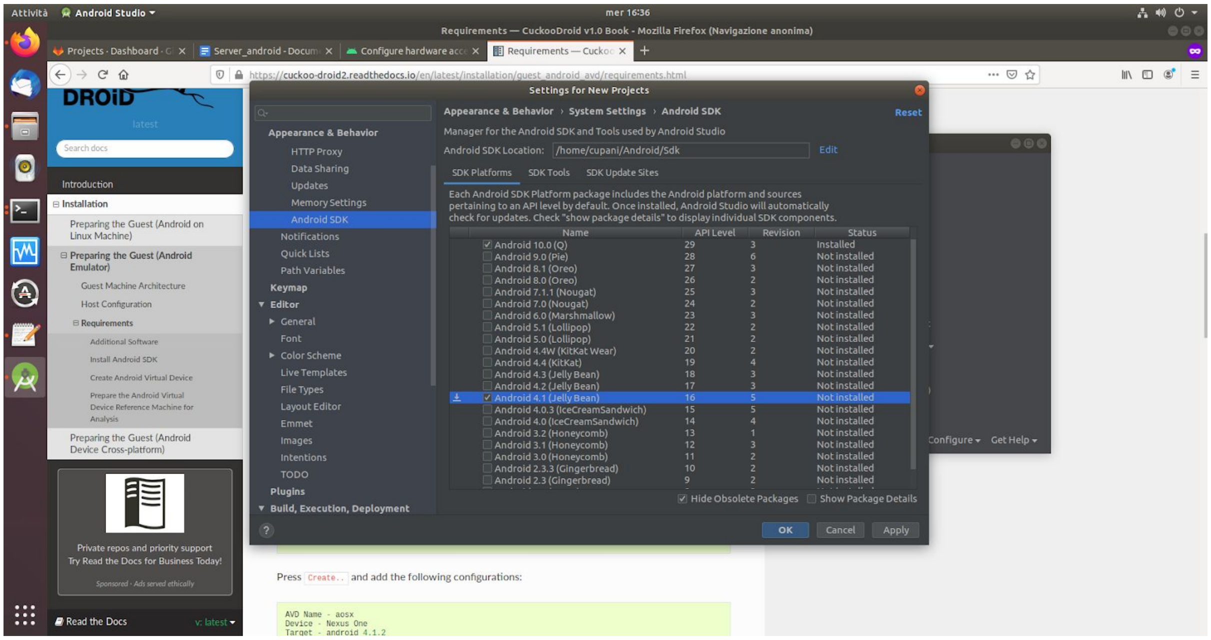
Changed in version 2.0-rc2: We used to suggest Windows XP as a guest VM but nowadays a 64-bit Windows 7 machine yields much better results.

KVM Users - Be sure to choose a hard drive image format that supports snapshots. See Saving

#To make Cuckoo run properly with the Android Emulator, install these required software and libraries on the Cuckoo host.
sudo apt-get install libstdc++6:i386 libgcc1:i386 zlib1g:i386 libncurses5:i386
sudo apt-get install openjdk-8-jre

#grant the current user the permission to use KVM
sudo adduser $USER kvm # $USER is the user that will run cuckoo, pay attention if you are running this command in a root shell!!

install Android 4.1(API 16) package from SDK manager



Settings for New Projects

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: /home/cupani/Android/Sdk

SDK Platforms | SDK Tools | SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

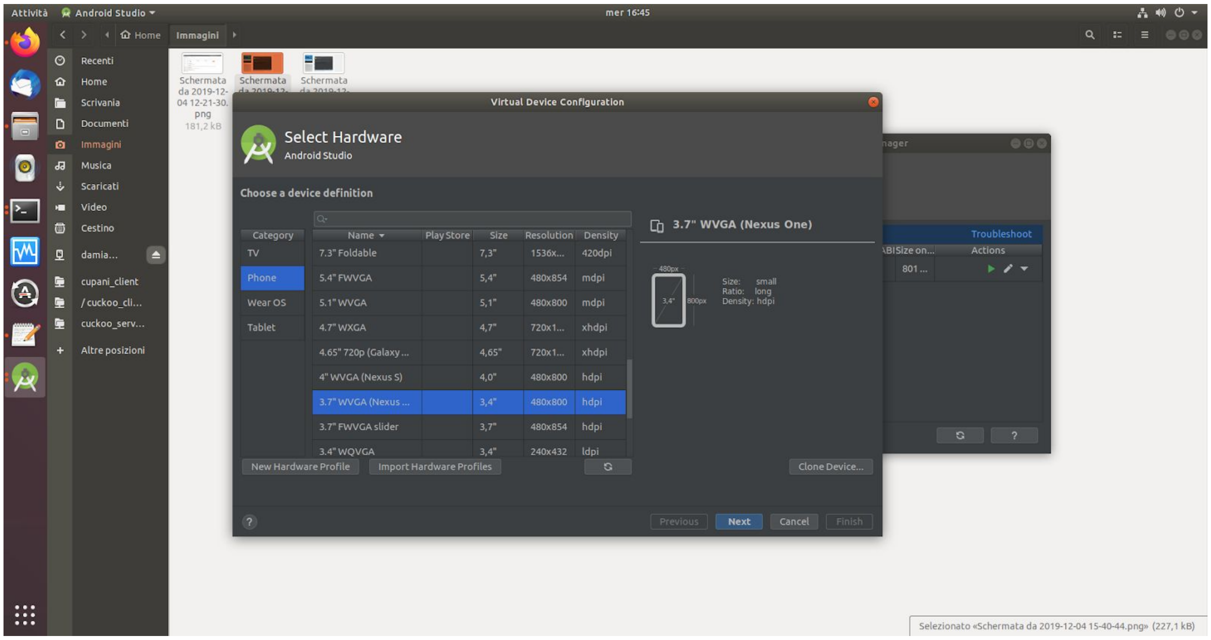
Name	API Level	Revision	Status
☑ Android 10.0 (Q)	29	3	Installed
□ Android 9.0 (Pie)	28	6	Not installed
□ Android 8.1 (Oreo)	27	3	Not installed
□ Android 8.0 (Oreo)	26	2	Not installed
□ Android 7.1.1 (Nougat)	25	3	Not installed
□ Android 7.0 (Nougat)	24	2	Not installed
□ Android 6.0 (Marshmallow)	23	3	Not installed
□ Android 5.1 (Lollipop)	22	2	Not installed
□ Android 5.0 (Lollipop)	21	2	Not installed
□ Android 4.4W (KitKat Wear)	20	2	Not installed
□ Android 4.4 (KitKat)	19	4	Not installed
□ Android 4.3 (Jelly Bean)	18	3	Not installed
□ Android 4.2 (Jelly Bean)	17	3	Not installed
☑ Android 4.1 (Jelly Bean)	16	5	Not installed
□ Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
□ Android 4.0 (IceCreamSandwich)	14	4	Not installed
□ Android 3.2 (Honeycomb)	13	1	Not installed
□ Android 3.1 (Honeycomb)	12	3	Not installed
□ Android 3.0 (Honeycomb)	11	2	Not installed
□ Android 2.3.3 (Gingerbread)	10	2	Not installed
□ Android 2.3 (Gingerbread)	9	2	Not installed

☑ Hide Obsolete Packages □ Show Package Details

Requirements — CuckooDroid v1.0 Book - Mozilla Firefox (Navigazione anonima)
https://cuckoo-droid2.readthedocs.io/en/latest/installation/guest_android_avd/requirements.html
Appearance & Behavior
HTTP Proxy
Data Sharing
Updates
Memory Settings
Android SDK
Notifications
Quick Lists
Path Variables
Keymap
Editor
General
Font
Color Scheme
Live Templates
File Types
Layout Editor
Emmet
Images
Intentions
TODO
Plugins
Build, Execution, Deployment
Reset
Edit
OK
Cancel
Apply
Introduction
Installation
Preparing the Guest (Android on Linux Machine)
Preparing the Guest (Android Emulator)
Guest Machine Architecture
Host Configuration
Requirements
Additional Software
Install Android SDK
Create Android Virtual Device
Prepare the Android Virtual Device Reference Machine for Analysis
Preparing the Guest (Android Device Cross-platform)
Press Create.. and add the following configurations:
AVD Name - aosx
Device - Nexus One
Target - android 4.1.2

#add Android SDK executables to $PATH if they are not already present
export PATH=$PATH:~/Android/Sdk/emulator::~/Android/Sdk/tools:~/Android/Sdk/build-tools/29.0.2/:~/Android/Sdk/platform-tools

create an Android Virtual Device following this guide
https://web.archive.org/web/20191205093116/https://cuckoo-droid2.readthedocs.io/en/latest/installation/guest_android_avd/requirements.html



Virtual Device Configuration

Select Hardware
Android Studio

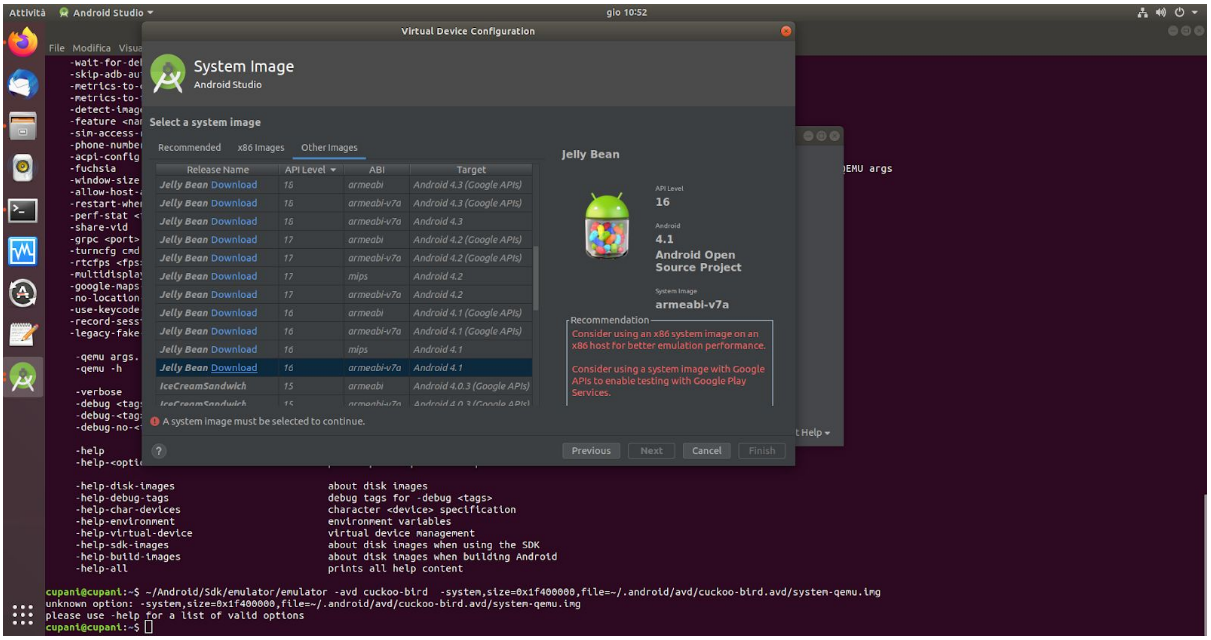
Choose a device definition

Category	Name	Play Store	Size	Resolution	Density
TV	7.3" Foldable		7,3"	1536x...	420dpi
Phone	5.4" FWVGA		5,4"	480x854	mdpi
Wear OS	5.1" WVGA		5,1"	480x800	mdpi
Tablet	4.7" WXGA		4,7"	720x1...	xhdpi
	4.65" 720p (Galaxy ...		4,65"	720x1...	xhdpi
	4" WVGA (Nexus S)		4,0"	480x800	hdpi
	3.7" WVGA (Nexus ...		3,4"	480x800	hdpi
	3.7" FWVGA slider		3,7"	480x854	hdpi
	3.4" WQVGA		3,4"	240x432	ldpi

3.7" WVGA (Nexus One)

Size: small
Ratio: long
Density: hdpi

in order to do this, download the arm
 android 4.1 level 16 API without Google Play Services(because the Google play services image cannot be rooted) image:



Virtual Device Configuration

System Image
Android Studio

Select a system image

Release Name	API Level	ABI	Target
Jelly Bean Download	18	armeabi	Android 4.3 (Google APIs)
Jelly Bean Download	18	armeabi-v7a	Android 4.3 (Google APIs)
Jelly Bean Download	18	armeabi-v7a	Android 4.3
Jelly Bean Download	17	armeabi	Android 4.2 (Google APIs)
Jelly Bean Download	17	armeabi-v7a	Android 4.2 (Google APIs)
Jelly Bean Download	17	mips	Android 4.2
Jelly Bean Download	17	armeabi-v7a	Android 4.2
Jelly Bean Download	16	armeabi	Android 4.1 (Google APIs)
Jelly Bean Download	16	armeabi-v7a	Android 4.1 (Google APIs)
Jelly Bean Download	16	mips	Android 4.1
Jelly Bean Download	16	armeabi-v7a	Android 4.1
IceCreamSandwich	15	armeabi	Android 4.0.3 (Google APIs)
IceCreamSandwich	15	armeabi-v7a	Android 4.0.3 (Google APIs)

Jelly Bean

API Level
16

Android
4.1
Android Open Source Project

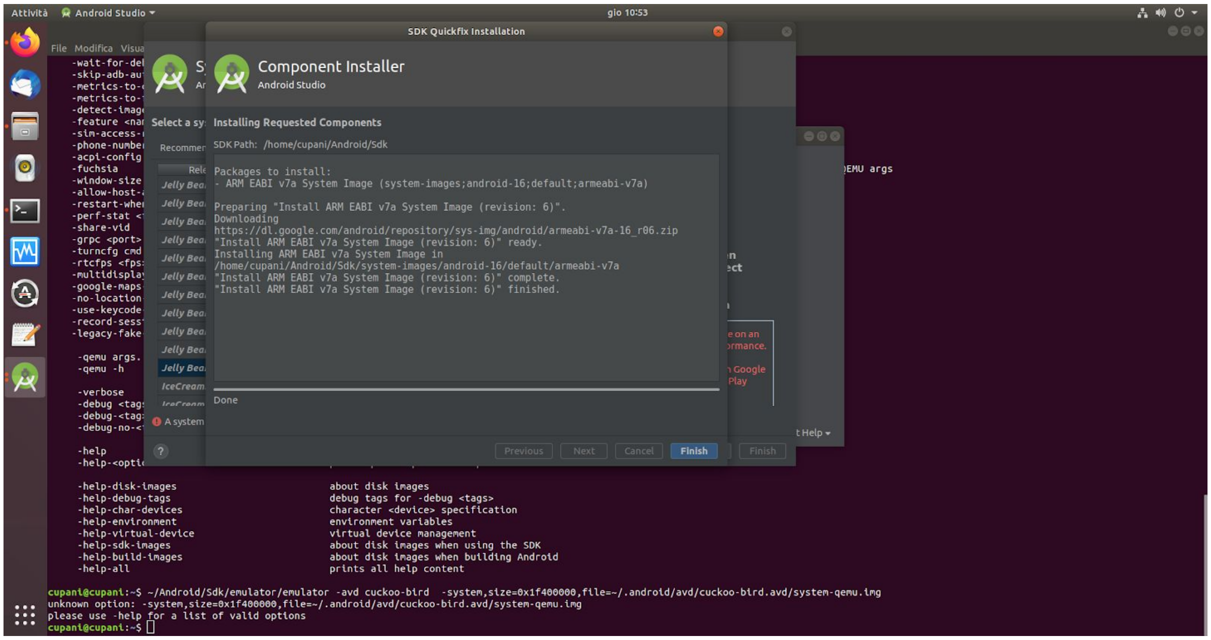
System Image
armeabi-v7a

Recommendation
Consider using an x86 system image on an x86 host for better emulation performance.

Consider using a system image with Google APIs to enable testing with Google Play Services.

A system image must be selected to continue.

```
   -wait-for-del
   -skip-adb-au
   -metrics-to-
   -metrics-to-
   -detect-imag
   -feature <na
   -sim-access-
   -phone-numbe
   -acpi-config
   -fuchsia
   -window-size
   -allow-host-
   -restart-whe
   -perf-stat <
   -share-vid
   -grpc <port>
   -turncfg cmd
   -rtcfps <fps
   -multidispla
   -google-maps
   -no-location
   -use-keycode
   -record-sess
   -legacy-fake

    -qemu args.
    -qemu -h

    -verbose
    -debug <tag
    -debug-<tag
    -debug-no-<

    -help
    -help-<opti

    -help-disk-images                         about disk images
    -help-debug-tags                          debug tags for -debug <tags>
    -help-char-devices                        character <device> specification
    -help-environment                         environment variables
    -help-virtual-device                      virtual device management
    -help-sdk-images                          about disk images when using the SDK
    -help-build-images                        about disk images when building Android
    -help-all                                 prints all help content

cupani@cupani:~$ ~/Android/Sdk/emulator/emulator -avd cuckoo-bird  -system,size=0x1f400000,file=~/.android/avd/cuckoo-bird.avd/system-qemu.img
unknown option: -system,size=0x1f400000,file=~/.android/avd/cuckoo-bird.avd/system-qemu.img
please use -help for a list of valid options
cupani@cupani:~$ 
```



Installing Requested Components

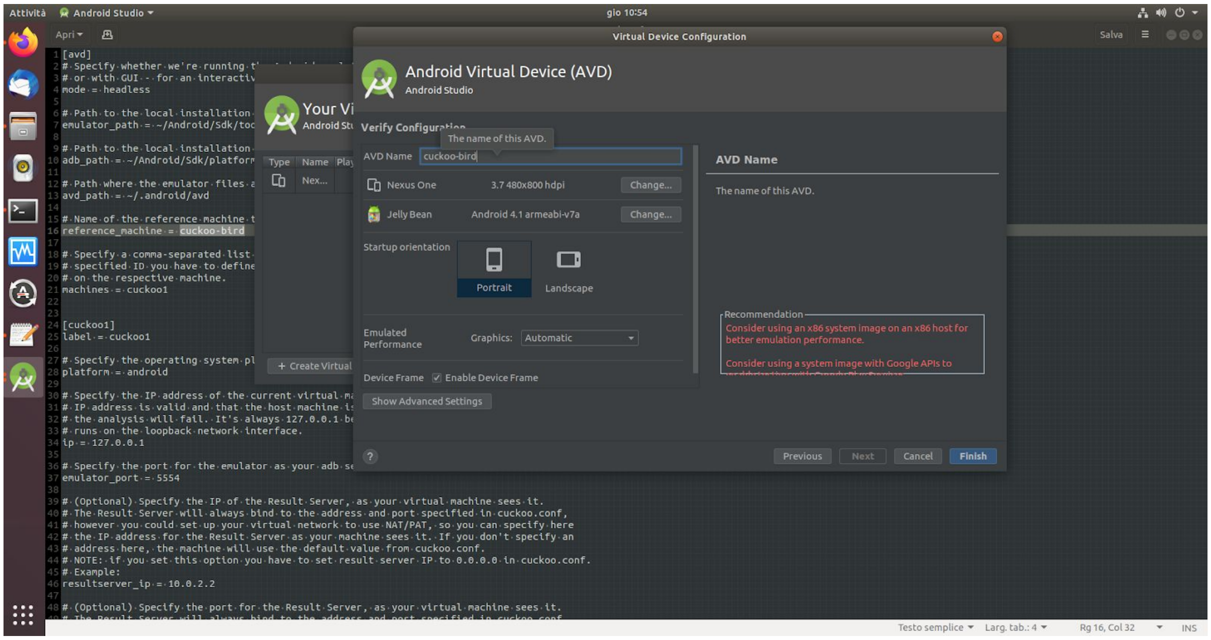
SDK Path: /home/cupani/Android/Sdk

```
Packages to install:
- ARM EABI v7a System Image (system-images;android-16;default;armeabi-v7a)

Preparing "Install ARM EABI v7a System Image (revision: 6)".
Downloading
https://dl.google.com/android/repository/sys-img/android/armeabi-v7a-16_r06.zip
"Install ARM EABI v7a System Image (revision: 6)" ready.
Installing ARM EABI v7a System Image in
/home/cupani/Android/Sdk/system-images/android-16/default/armeabi-v7a
"Install ARM EABI v7a System Image (revision: 6)" complete.
"Install ARM EABI v7a System Image (revision: 6)" finished.
```

Done

```
    -help-disk-images                          about disk images
    -help-debug-tags                           debug tags for -debug <tags>
    -help-char-devices                         character <device> specification
    -help-environment                          environment variables
    -help-virtual-device                       virtual device management
    -help-sdk-images                           about disk images when using the SDK
    -help-build-images                         about disk images when building Android
    -help-all                                  prints all help content

cupani@cupani:~$ ~/Android/Sdk/emulator/emulator -avd cuckoo-bird  -system,size=0x1f400000,file=~/.android/avd/cuckoo-bird.avd/system-qemu.img
unknown option: -system,size=0x1f400000,file=~/.android/avd/cuckoo-bird.avd/system-qemu.img
please use -help for a list of valid options
cupani@cupani:~$ 
```

choose the AVD name so it match the value written in avd.conf(in this case cuckoo-bird)



```
[avd]
# Specify whether we're running t
# or with GUI - for an interactiv
mode = headless

# Path to the local installation
emulator_path = ~/Android/Sdk/too

# Path to the local installation
adb_path = ~/Android/Sdk/platform

# Path where the emulator files a
avd_path = ~/.android/avd

# Name of the reference machine t
reference_machine = cuckoo-bird

# Specify a comma-separated list
# specified ID you have to define
# on the respective machine.
machines = cuckoo1


[cuckoo1]
label = cuckoo1

# Specify the operating system pl
platform = android

# Specify the IP address of the current virtual ma
# IP address is valid and that the host machine is
# the analysis will fail. It's always 127.0.0.1 be
# runs on the loopback network interface.
ip = 127.0.0.1

# Specify the port for the emulator as your adb se
emulator_port = 5554

# (Optional) Specify the IP of the Result Server, as your virtual machine sees it.
# The Result Server will always bind to the address and port specified in cuckoo.conf,
# however you could set up your virtual network to use NAT/PAT, so you can specify here
# the IP address for the Result Server as your machine sees it. If you don't specify an
# address here, the machine will use the default value from cuckoo.conf.
# NOTE: if you set this option you have to set result server IP to 0.0.0.0 in cuckoo.conf.
# Example:
resultserver_ip = 10.0.2.2

# (Optional) Specify the port for the Result Server, as your virtual machine sees it.
```

AVD Name: cuckoo-bird

Nexus One 3.7 480x800 hdpi Change...

Jelly Bean Android 4.1 armeabi-v7a Change...

Emulated Performance Graphics: Automatic

Device Frame ☑ Enable Device Frame

AVD Name

The name of this AVD.

Recommendation

Consider using an x86 system image on an x86 host for better emulation performance.

Consider using a system image with Google APIs to

Before starting the emulator do the following:
cp /home/cupani/Android/Sdk/system-images/android-16/default/armeabi-v7a/system.img ~/.android/avd/cuckoo-bird.avd/system-qemu.img

After that starts the emulator with the following command:

```
~/Android/Sdk/emulator/emulator -avd cuckoo-bird -writable-system  -system ~/.android/avd/cuckoo-bird.avd/system-qemu.img -qemu
```

After the emulator finished the boot run the following script :
“/home/USER/cuckoo/utils/android_emulator_creator/create_guest_avd.sh” but only after replacing the 47th line with “ $ADB push ../../agent/android/python_agent/* /data/local/ “
e attendere che vengano installati tutte le componenti all’interno dell’AVD.
[vedi
https://github.com/damianocupani/cuckoo-droid/commit/d180278b8d36b6b84c7e84230cbceb7e2207b9a1]

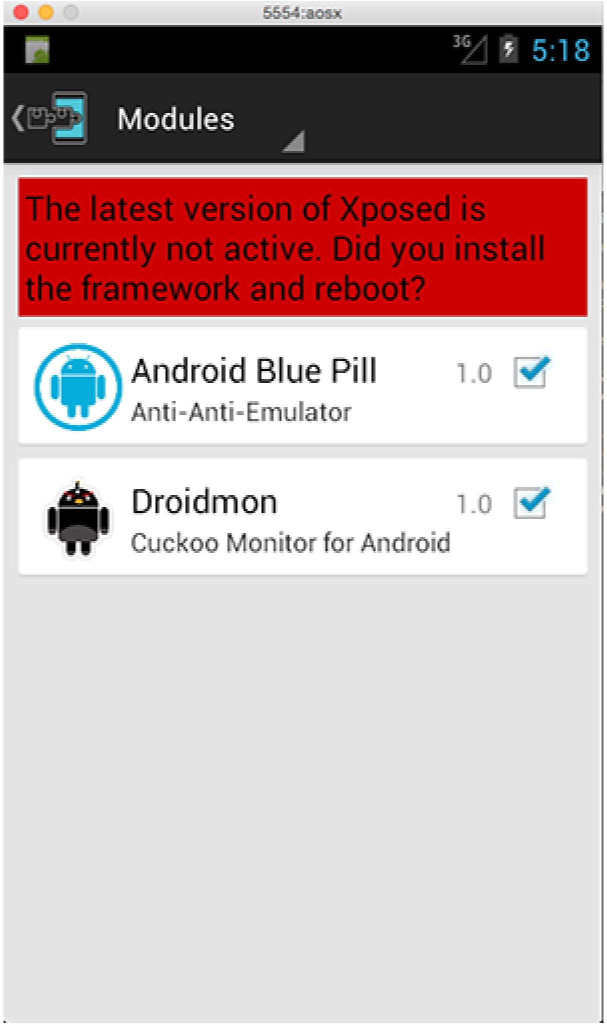
to resolve the” cannot connet to adb on port 5037” error, you can try starting first an AVD from the GUI and then check if ABD is running with the following command:
sudo netstat -tulpn | grep 5037 or
sudo netstat -tulpn | grep adb

to solve the “failed to create drawable” error see:
https://web.archive.org/web/20191210143141/https://stackoverflow.com/questions/35182518/android-studio-failed-to-create-drawable

Rooting AVD (Android Virtual Device)

Inside the Virtual Device do the following:

- Press settings->security->screenlock->none
  Press settings->Display->sleep->30 minutes
- Start Generate contacts app
- Start Supersuser app
- Start xposedinstaller app
- In Modules, check both packages Droidmon , Android Blue Pill
- Press framework -> install -> cancel-> soft reboot



Modules

The latest version of Xposed is currently not active. Did you install the framework and reboot?

Android Blue Pill 1.0 ☑
Anti-Anti-Emulator

Droidmon 1.0 ☑
Cuckoo Monitor for Android

Bug fixes

- edit the file “/home/USER/cuckoo/analyzer/android/lib/api/adb.py” , more specifically the

function *execute_sample* on 111th line with the following:
proc =

subprocess.Popen("/system/bin/am start -n"+ package+"/"+activity,

stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True,

executable="/system/bin/sh") e commentare la riga sottostante.

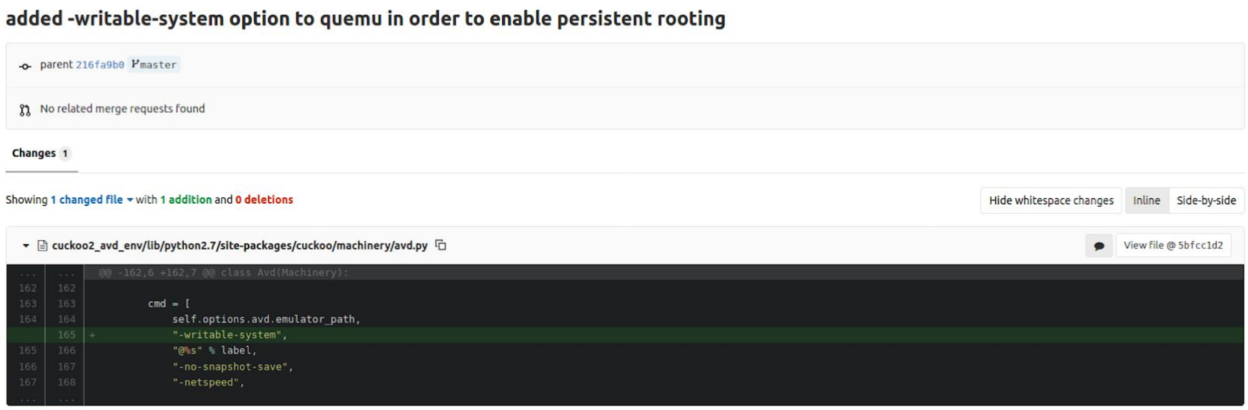
- edit the file “/home/USER/cuckoo/modules/processing/network.py” , more specifically replace the 596th line with: **results = Pcap(self.pcap_path).run()**

- edit the file “/home/USER/cuckooTest/cuckoo/modules/reporting/mongodb.py” , more specifically, replace the last line with: **self.conn.close()**

With the recents version of AVD another fix is needed in order to make the persistent rooting working

Edit the file cuckoo/machinery/avd.py as shown in the following:

https://github.com/damianocupani/my_cuckoodroid/commit/b06a3b27bf8b75ef83ea0ce10fb7de3a4c667bb2



added -writable-system option to quemu in order to enable persistent rooting

parent 216fa9b0 master

No related merge requests found

Showing 1 changed file with 1 addition and 0 deletions

cuckoo2_avd_env/lib/python2.7/site-packages/cuckoo/machinery/avd.py

```
@@ -162,6 +162,7 @@ class Avd(Machinery):

        cmd = [
            self.options.avd.emulator_path,
+           "-writable-system",
            "@%s" % label,
            "-no-snapshot-save",
            "-netspeed",
```